Table of Contents


```
% NOM : LAANAIYA
% PRÉNOM : MAHMOUD

clear;
close all;
clc;
```

Bits

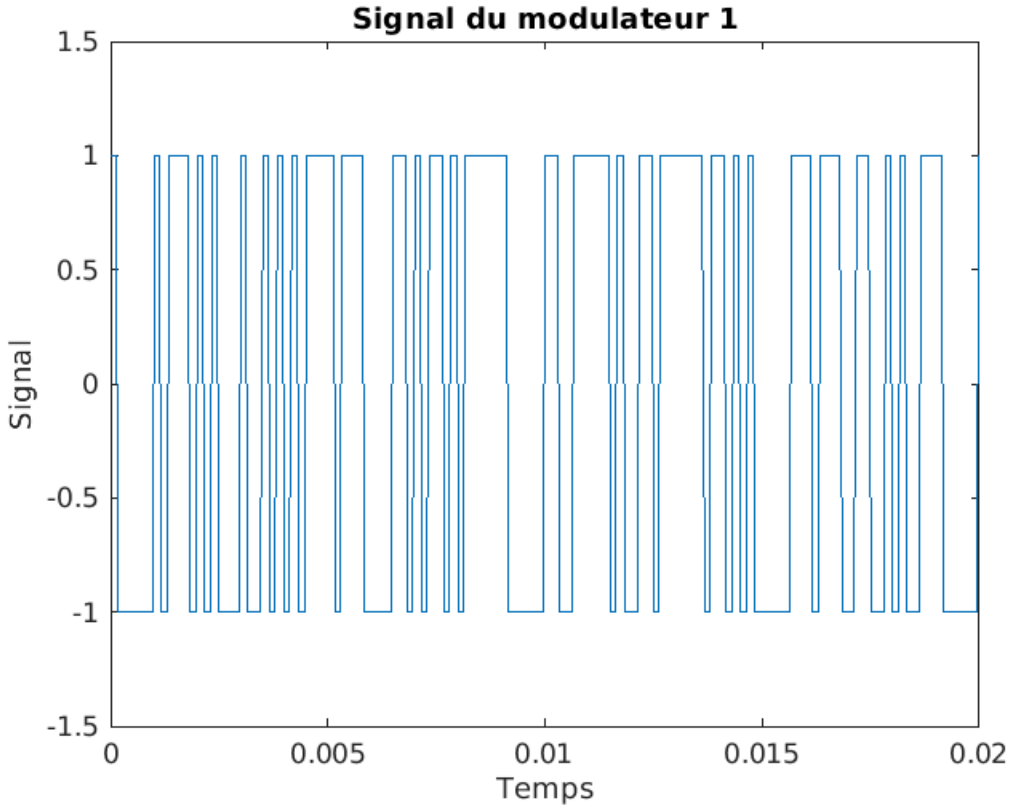
```
Donnees = randi([0,1], 1, 1000);
```

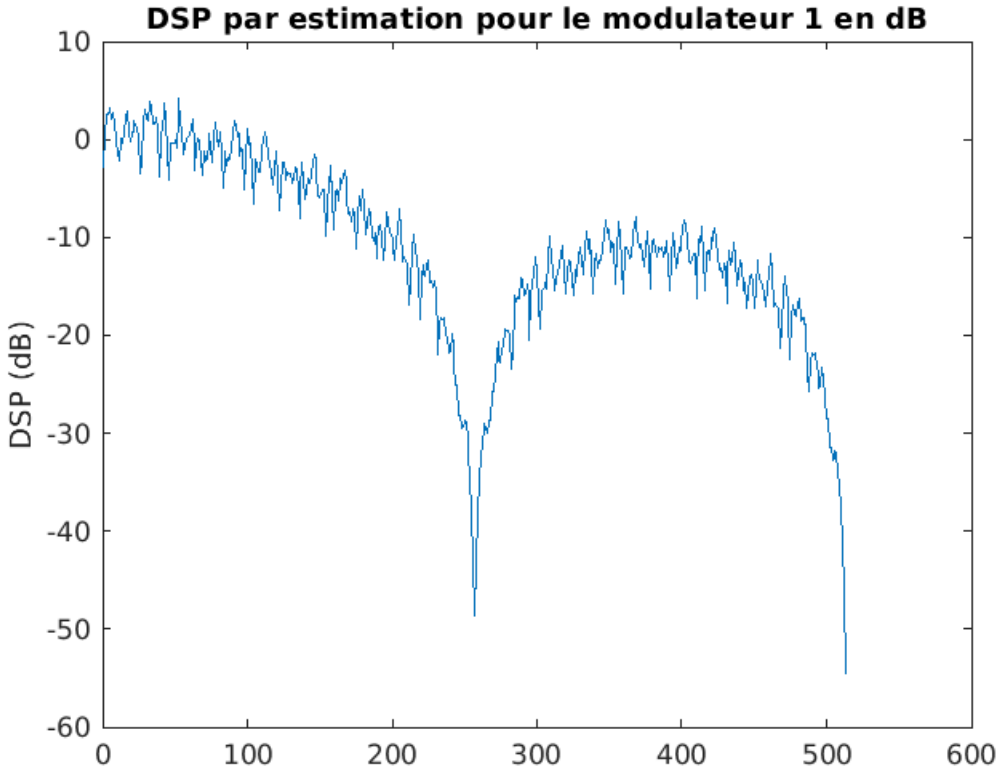
Modulateur 1

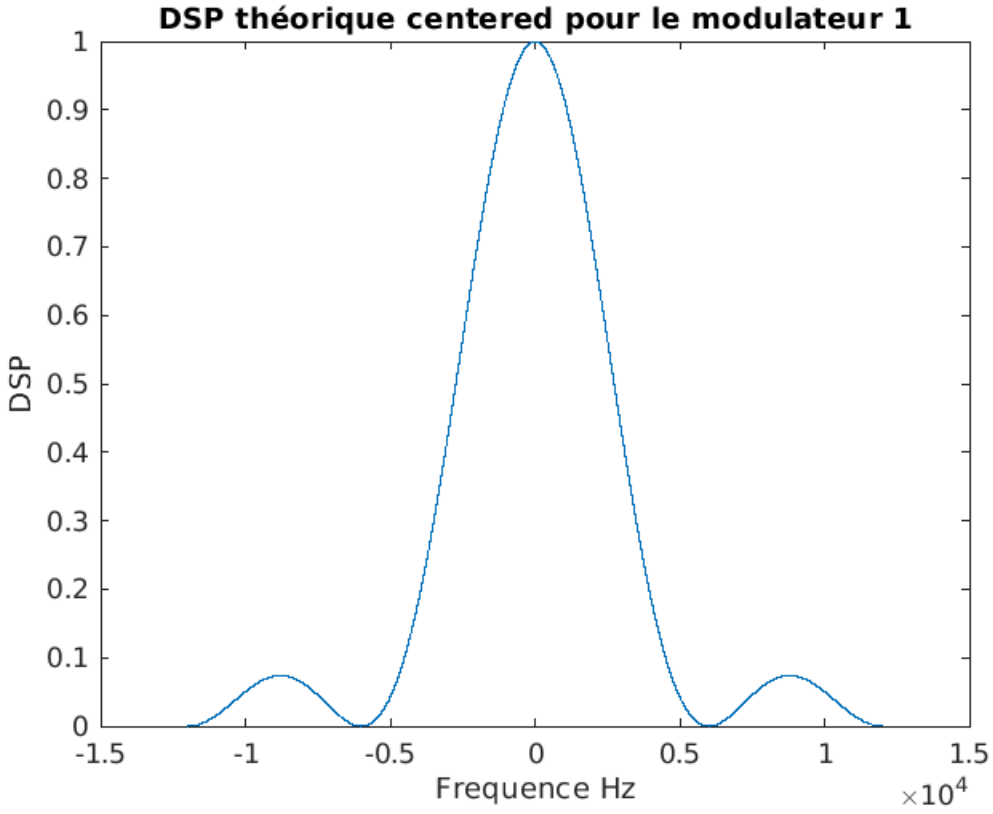
Mapping

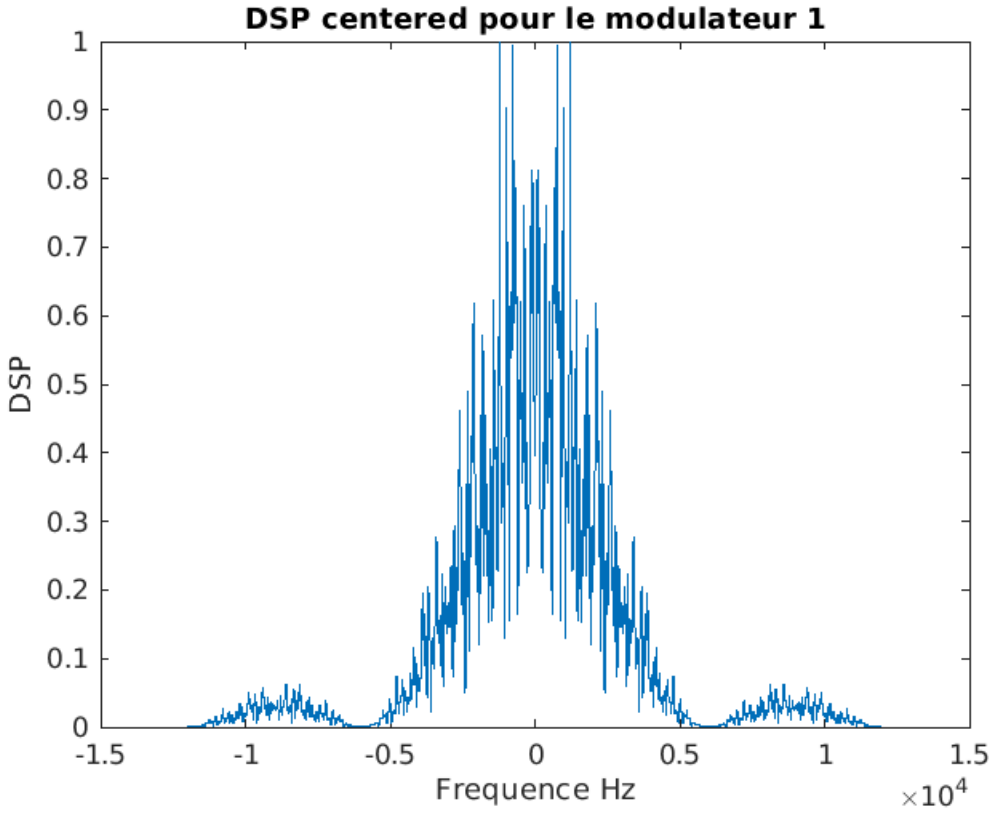
```
mapp1 = Donnees - (Donnees == 0);
% Suréchantillonage
Rb = 6000; % Débit binaire
M1 = 2; % M = 2^n (n=1)
Fe = 24000; % Fréquence d´échantillonage
Te = 1/Fe;
Rs = Rb/log2(M1); % Débit symbole
Ns1 = floor(Fe/Rs);
Ts1 = Ns1*Te;
h1 = ones(1, Ns1); % Filtre

% Etude par estimation
Kronr1 = kron(mapp1, [1 zeros(1, Ns1-1)]); % Positionnement des zéros
 entre les ak
signalfiltre1 = filter(h1, 1, Kronr1); % Filtrage
DSP_x1 = pwelch(signalfiltre1, [], [], Fe, "twosided"); % Densité
 spectrale de puissance
DSP_x1 = DSP_x1./max(DSP_x1); % Normaliser la DSP
DSP_x1_centered = pwelch(signalfiltre1, [], [], Fe, "centered"); %
 Densité spectrale de puissance centrée
DSP_x1_centered = DSP_x1_centered./max(DSP_x1_centered); % Normaliser
 la DSP centrée
DSP_x1_dB = 10*log10(pwelch(signalfiltre1)); % Densité spectrale de
 puissance en dB
```

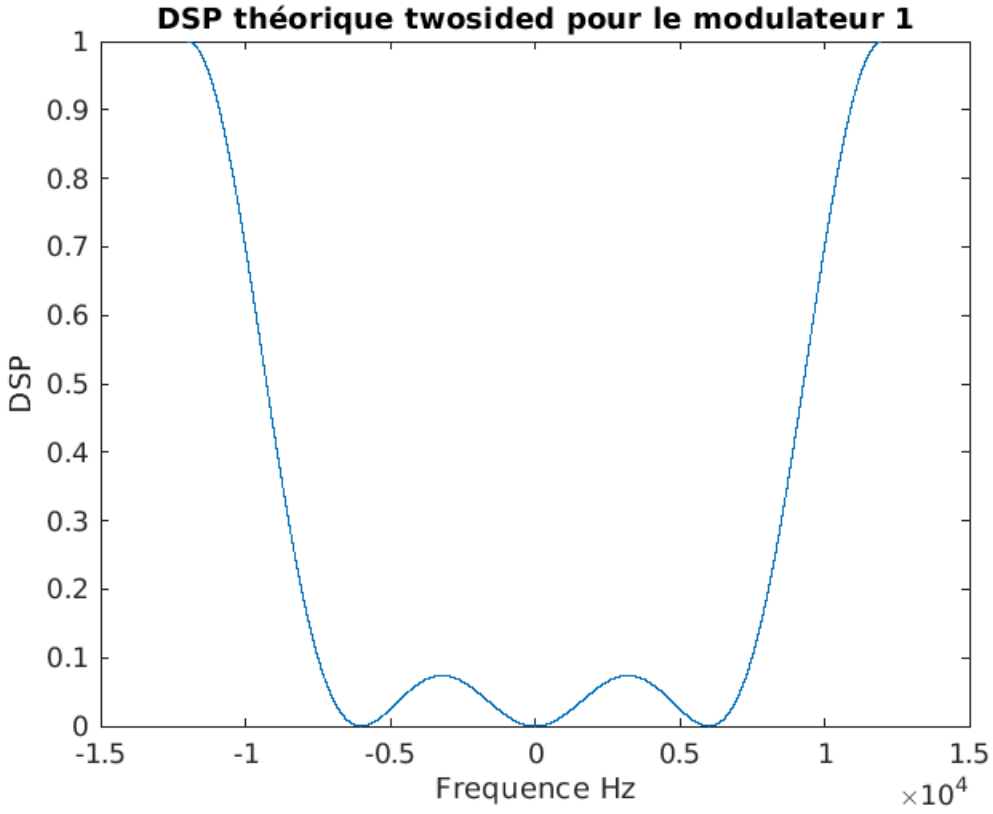
```
% Etude théorique
Tf1 = abs(fft(h1, Fe));
DSP_x1_theorique = (1/Ts1) * Tf1.^2; % Densité spectrale de puissance
 en théorique
DSP_x1_theorique = DSP_x1_theorique./max(DSP_x1_theorique); %
 Normaliser la DSP

% Tracés
t1 = 0:Te:(length(Kronr1)-1)*Te; % Interval de temps discret
figure,plot(t1,signalfiltre1);
axis([0 0.02 -1.5 1.5]);
xlabel("Temps");
ylabel("Signal");
title("Signal du modulateur 1");

frequence1 = -Fe/2:Fe/(length(DSP_x1)-1):Fe/2; % Interval de fréquence
 discret
figure,plot(DSP_x1_dB);
ylabel("DSP (dB)");
title("DSP par estimation pour le modulateur 1 en dB");
figure,plot(frequence1, fftshift(DSP_x1_theorique));
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP théorique centered pour le modulateur 1");
figure,plot(frequence1, DSP_x1_centered);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP centered pour le modulateur 1");
figure,plot(frequence1, DSP_x1_theorique);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP théorique twosided pour le modulateur 1");
figure,plot(frequence1, DSP_x1);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP par estimation twosided pour le modulateur 1");

% Comparaison DSPs Modulateur 1
figure();
title("Comparaison DSP Modulateur 1");
hold on
plot(frequence1, DSP_x1_centered, "DisplayName", "DSP-1-pratique");
plot(frequence1, fftshift(DSP_x1_theorique), "DisplayName", "DSP-1-
theorique" );
hold off
legend;
```

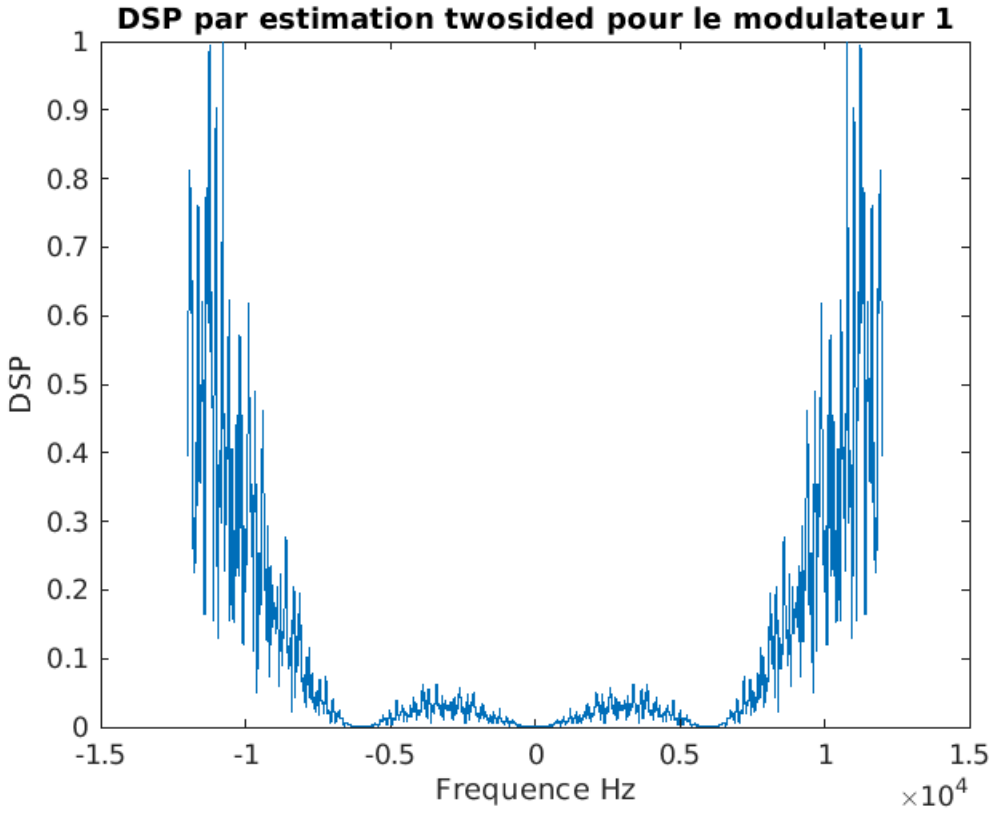

Signal du modulateur 1
1.5
1
0.5
0
-0.5
-1
-1.5
Signal
0
0.005
0.01
0.015
0.02
Temps

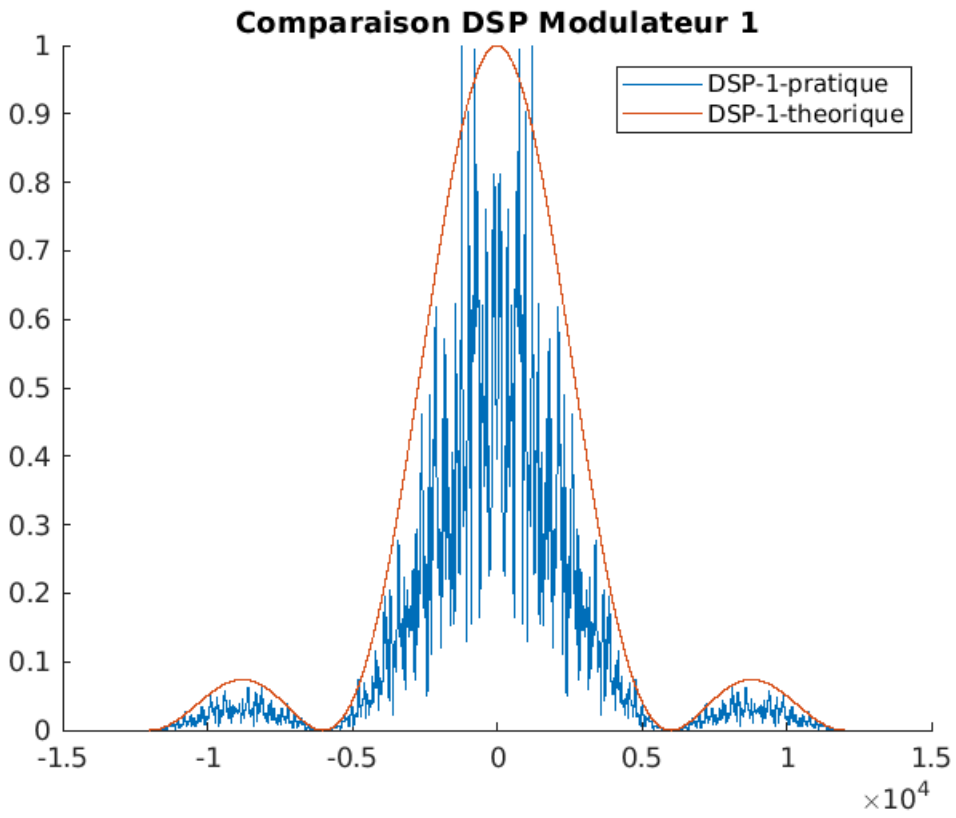

DSP par estimation pour le modulateur 1 en dB
10
0
-10
-20
-30
-40
-50
-60
DSP (dB)
0
100
200
300
400
500
600


DSP théorique centered pour le modulateur 1
DSP
Frequence Hz
×10^4


DSP centered pour le modulateur 1
DSP
Frequence Hz
×10^4


DSP théorique twosided pour le modulateur 1
DSP
Frequence Hz
×10^4


DSP par estimation twosided pour le modulateur 1
DSP
Frequence Hz
×10^4

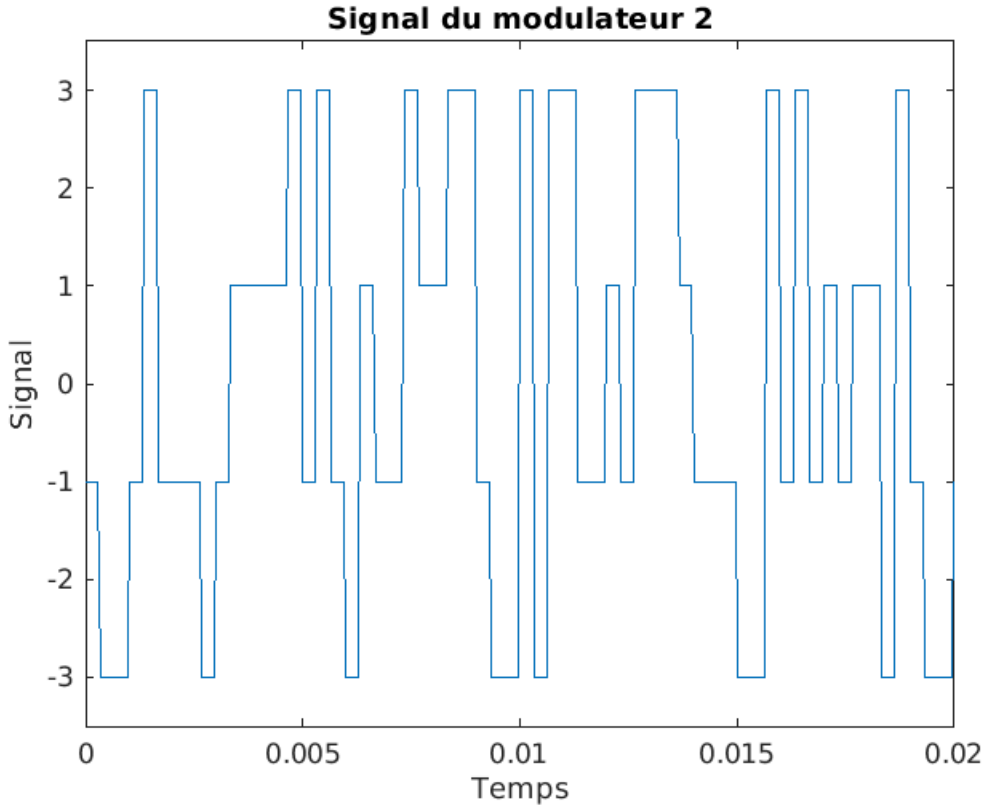


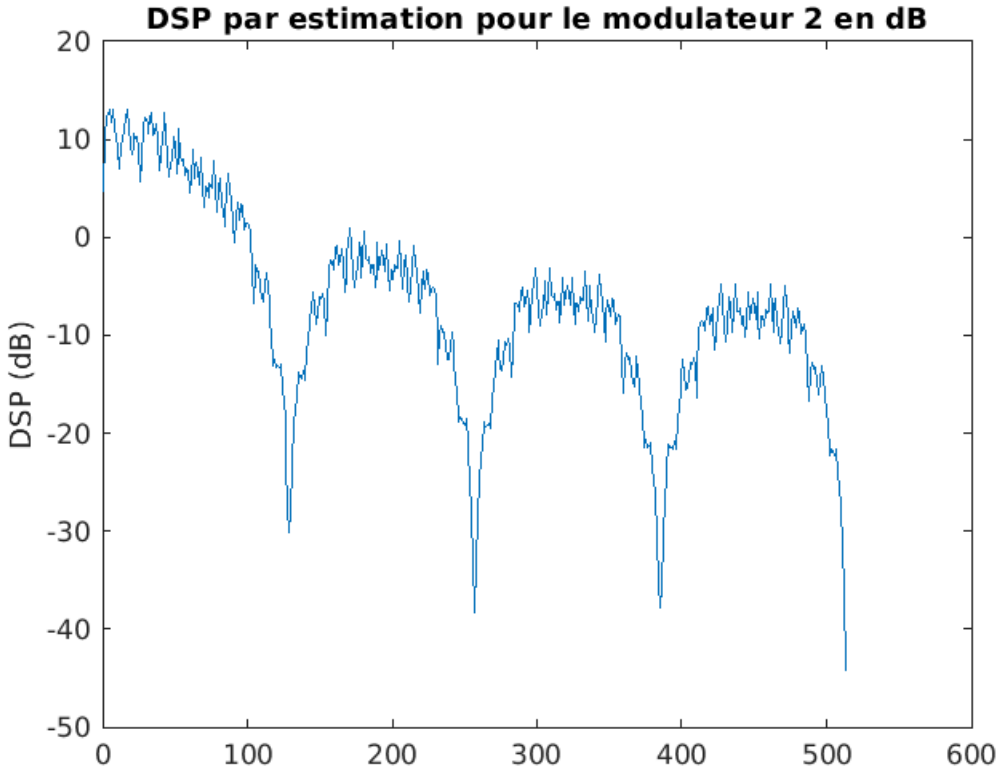
Modulateur 2

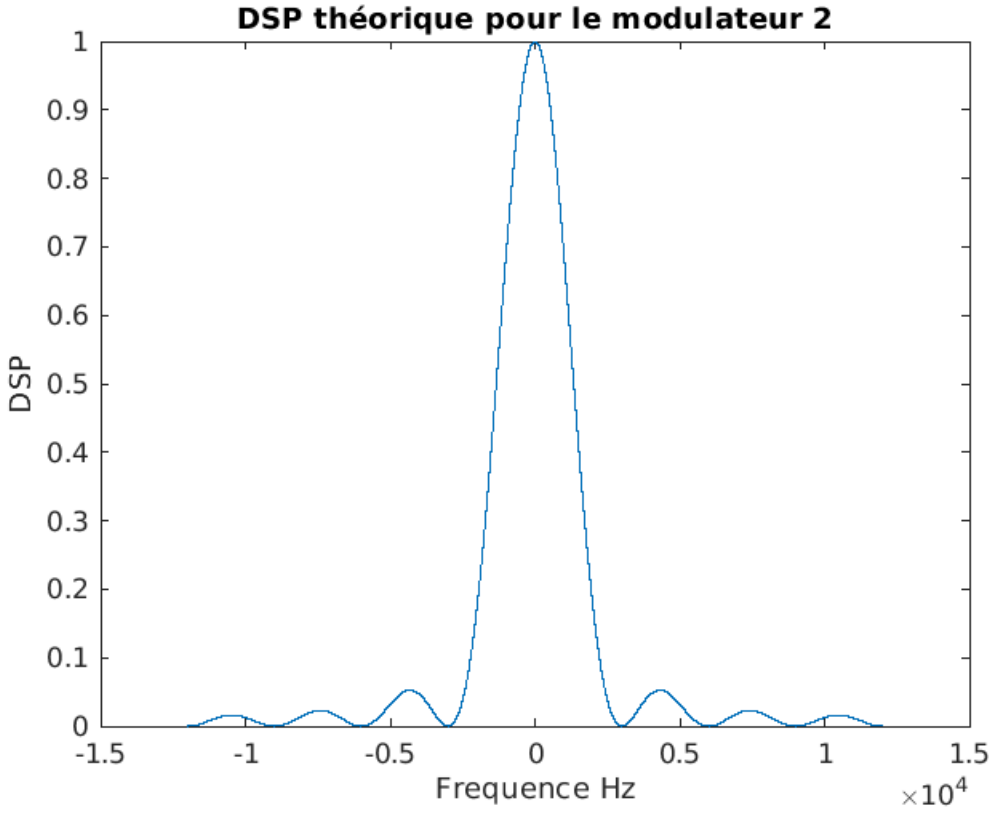
Mapping

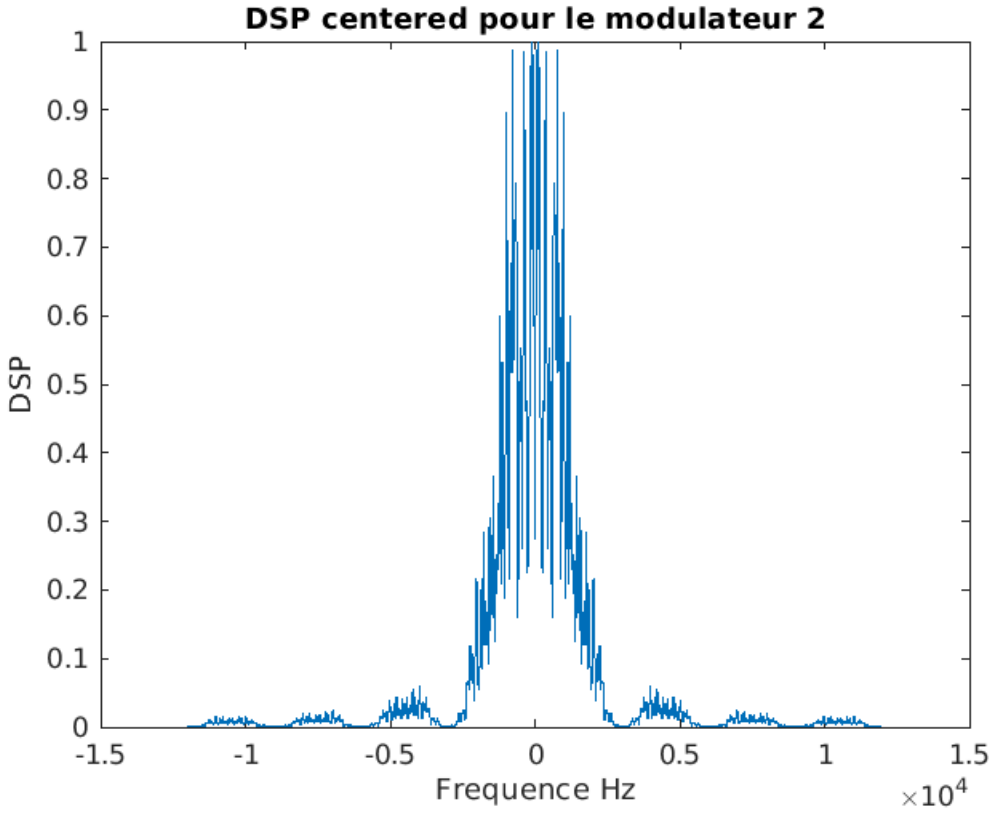
```
mapp2 = (2*bi2de(reshape(Donnees, 2, length(Donnees)/2).') - 3).'; %
 Du binaire au decimal
% Suréchantillonage
Ns2 = 8;
M2 = 4; % n = 2
h2 = ones(1,Ns2); % Filtre
Ts2 = Ns2*Te;

% Étude par estimation
Kronr2 = kron(mapp2, [1 zeros(1, Ns2-1)]); % Positionnement des zéros
 entre les ak
signalfiltre2 = filter(h2,1,Kronr2); % Filtrage
DSP_x2 = pwelch(signalfiltre2, [], [], Fe, "twosided"); % Densité
 spectrale de puissance
DSP_x2 = DSP_x2./max(DSP_x2); % Normaliser la DSP
DSP_x2_centered = pwelch(signalfiltre2, [], [], Fe, "centered"); %
 Densité spectrale de puissance centrée
DSP_x2_centered = DSP_x2_centered./max(DSP_x2_centered); % Normaliser
 la DSP centrée
DSP_x2_dB = 10*log10(pwelch(signalfiltre2)); % Densité spectrale de
 puissance en dB
```

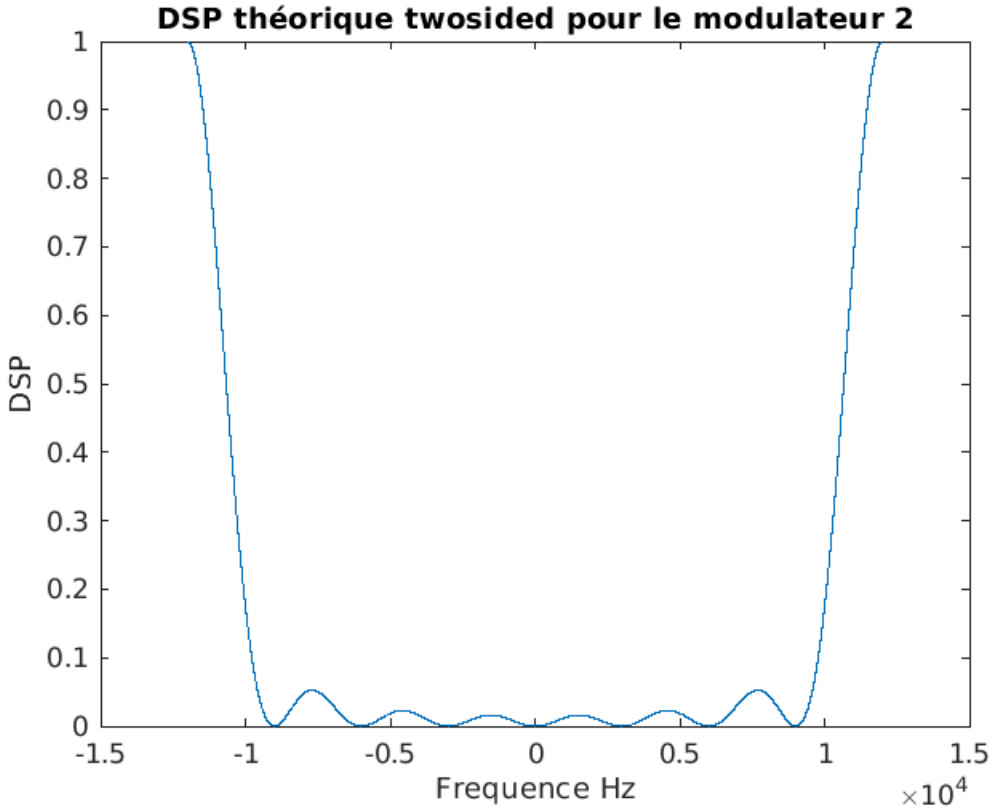
```
% Etude théorique
Tf2 = abs(fft(h2, Fe));
DSP_x2_theorique = (1/Ts2) * Tf2.^2; % Densité spectrale de puissance
 en théorique
DSP_x2_theorique = DSP_x2_theorique./max(DSP_x2_theorique); %
 Normaliser la DSP

% Tracés
t2 = 0:Te:(length(Kronr2)-1)*Te; % Interval de temps discret
figure,plot(t2,signalfiltre2);
axis([0 0.02 -3.5 3.5]);
xlabel("Temps");
ylabel("Signal");
title("Signal du modulateur 2");

frequence2 = -Fe/2:Fe/(length(DSP_x2)-1):Fe/2; % Interval de fréquence
 discret
figure,plot(DSP_x2_dB);
ylabel("DSP (dB)");
title("DSP par estimation pour le modulateur 2 en dB");
figure,plot(frequence2, fftshift(DSP_x2_theorique));
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP théorique pour le modulateur 2");
figure,plot(frequence2, DSP_x2_centered);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP centered pour le modulateur 2");
figure,plot(frequence2, DSP_x2_theorique);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP théorique twosided pour le modulateur 2");
figure,plot(frequence2, DSP_x2);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP par estimation twosided pour le modulateur 2");

% Comparaison DSPs Modulateur 2
figure();
title("Comparaison DSP Modulateur 2");
hold on
plot(frequence2, DSP_x2_centered, "DisplayName", "DSP-2-centered");
plot(frequence2, fftshift(DSP_x2_theorique), "DisplayName", "DSP-2-
theorique");
hold off
legend;
```

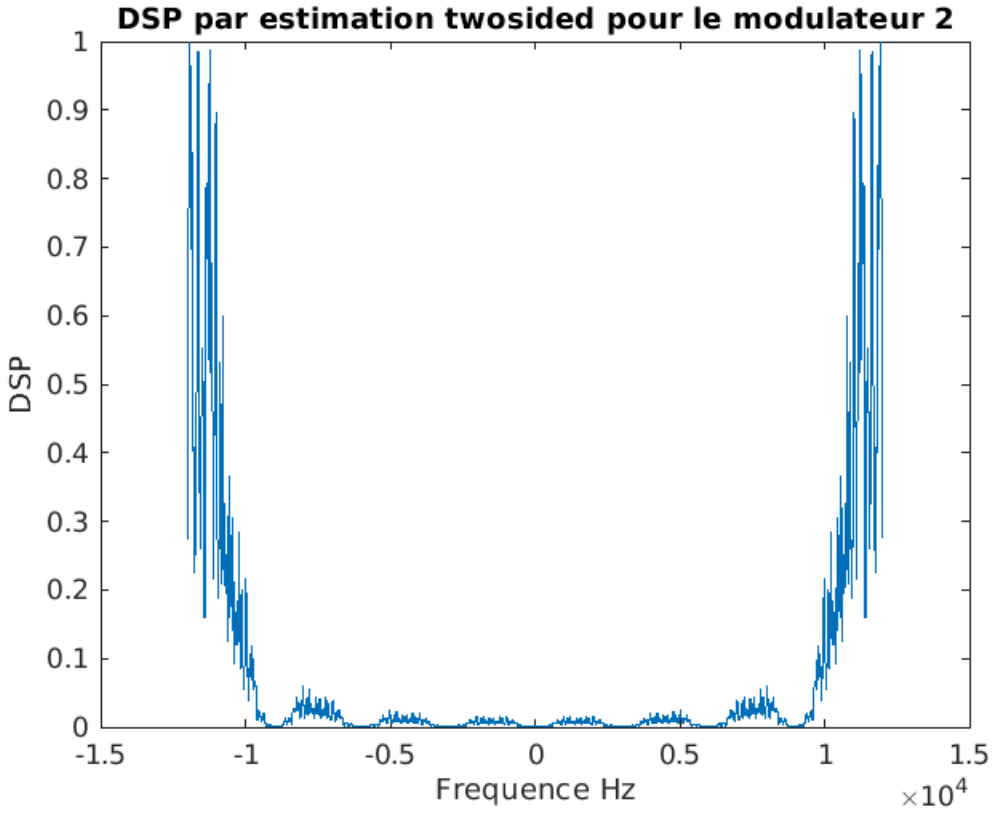

Signal du modulateur 2
Signal
Temps
3
2
1
0
-1
-2
-3
0
0.005
0.01
0.015
0.02

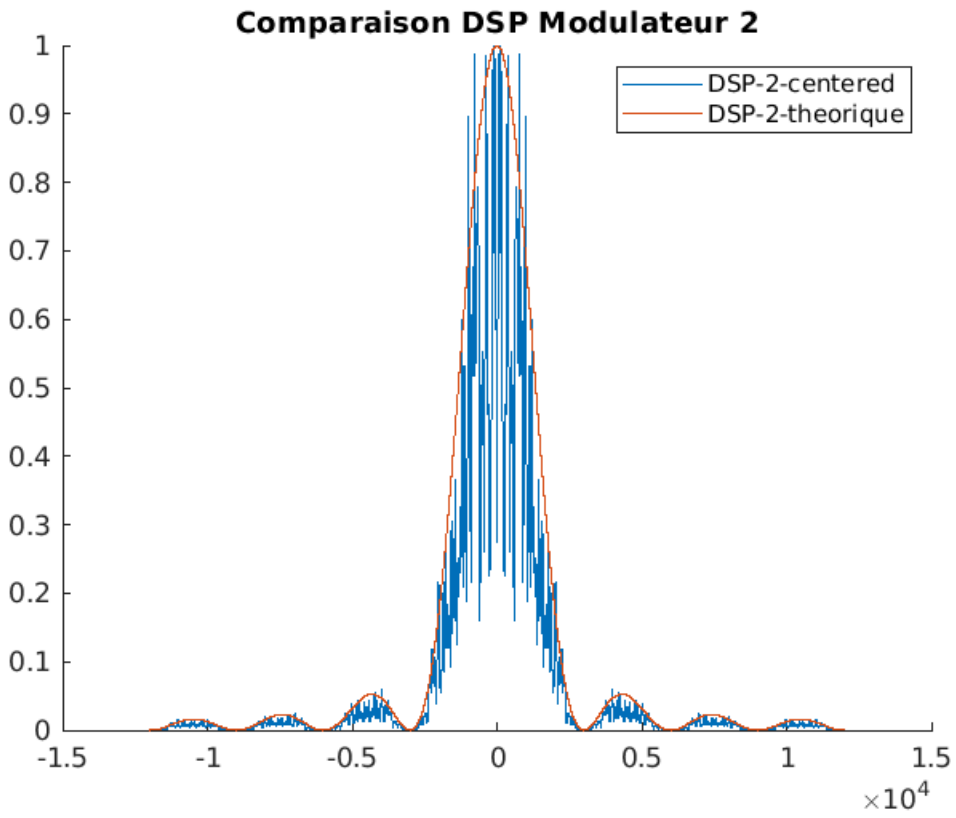

DSP par estimation pour le modulateur 2 en dB
DSP (dB)
20
10
0
-10
-20
-30
-40
-50
0
100
200
300
400
500
600


DSP théorique pour le modulateur 2
DSP
Frequence Hz
×10^4
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1.5
-1
-0.5
0
0.5
1
1.5


DSP centered pour le modulateur 2
DSP
Frequence Hz
×10^4
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1.5
-1
-0.5
0
0.5
1
1.5


DSP théorique twosided pour le modulateur 2
DSP
Frequence Hz
×10^4


DSP par estimation twosided pour le modulateur 2
DSP
Frequence Hz
×10^4

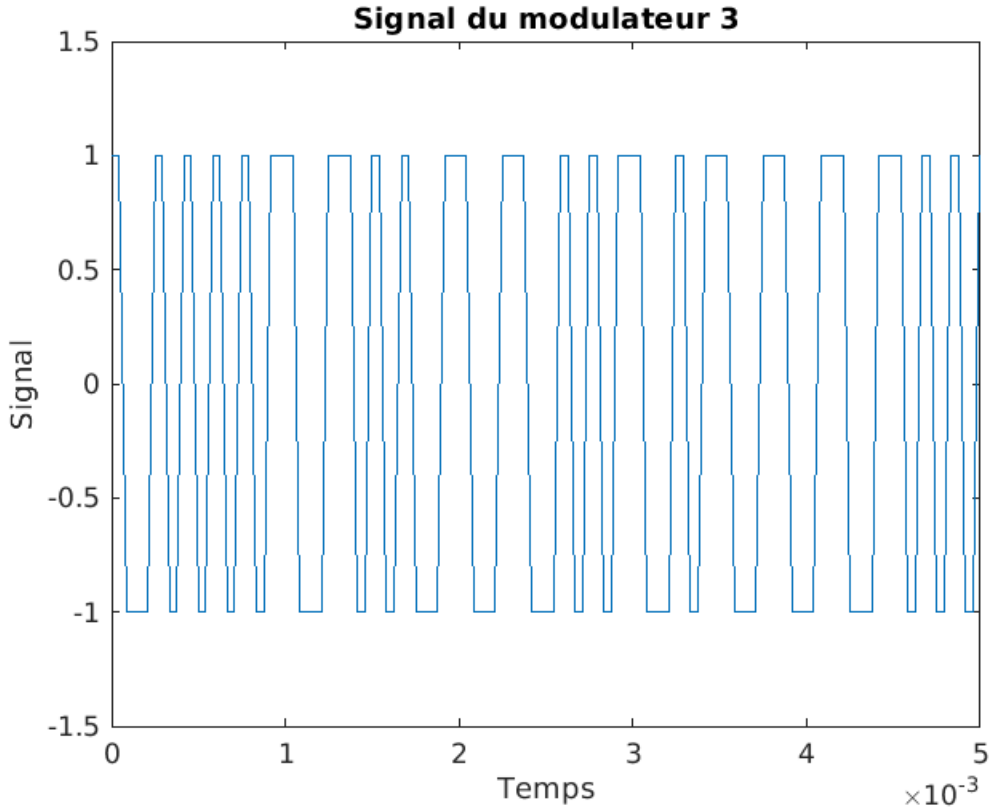


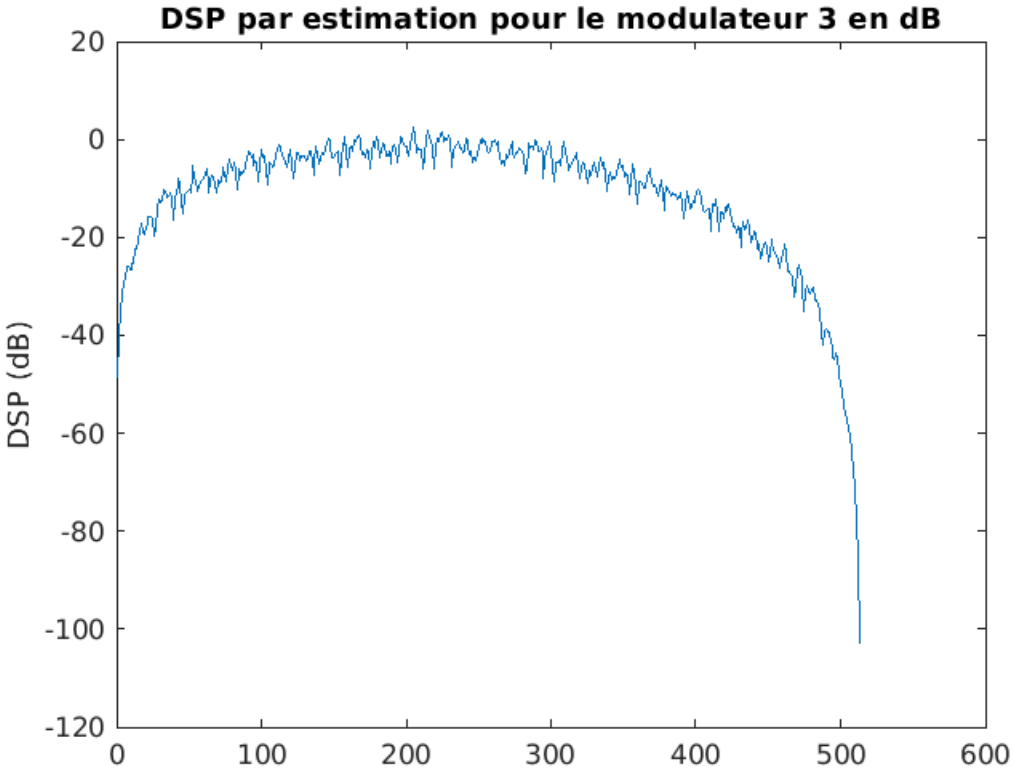
Modulateur 3

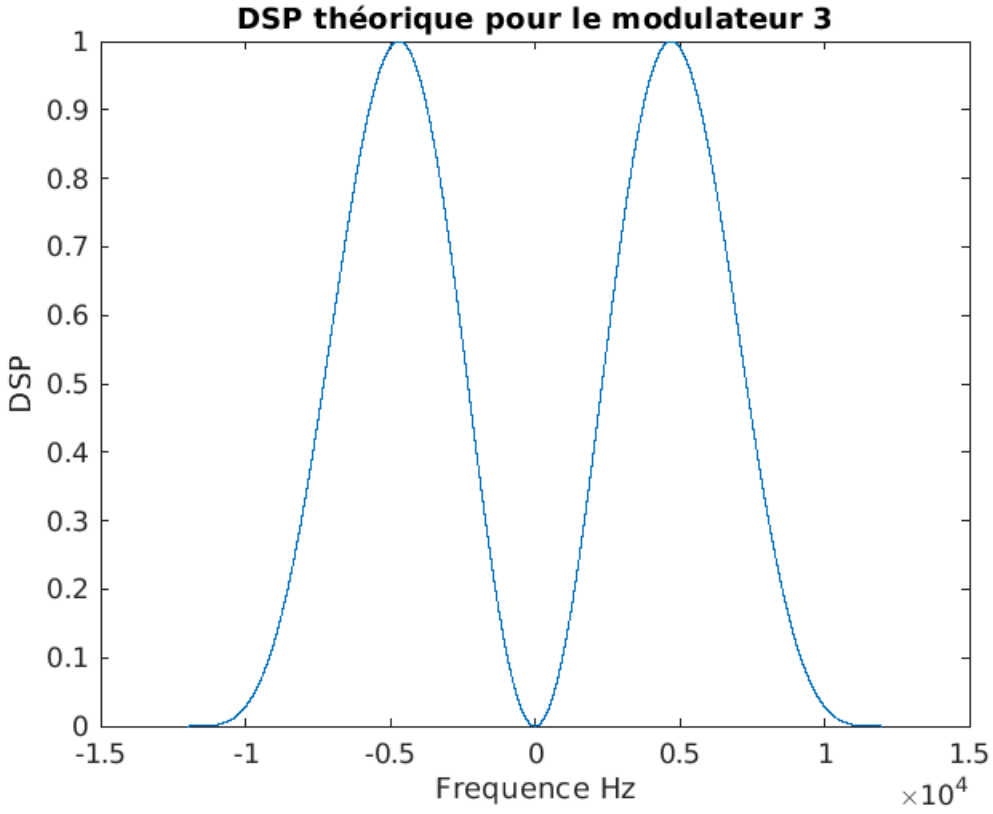
Mapping

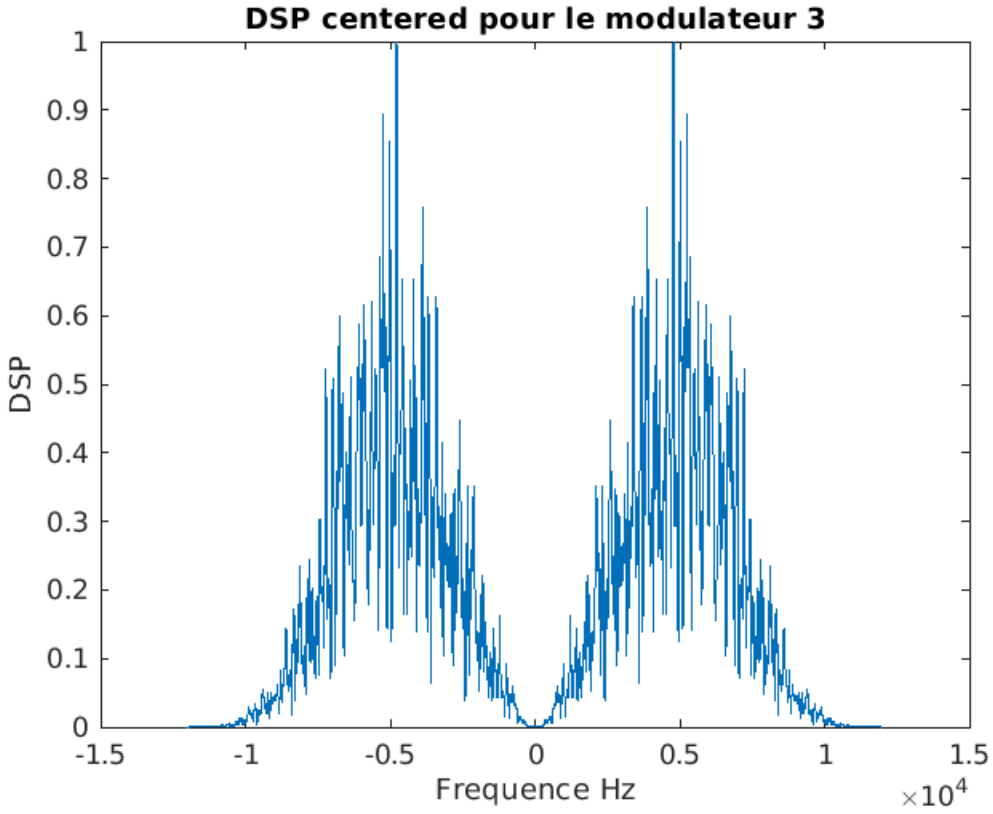
```
mapp3 = Donnees - (Donnees == 0);
% Suréchantillonage
h3 = [ones(1, Ns1/2) -1.*ones(1, Ns1/2)]; % Filtre front montant

% Étude par estimation
Kronr3 = kron(mapp3, [1 zeros(1, Ns1-1)]); % Positionnement des zéros
 entre les ak
signalfiltre3 = filter(h3, 1, Kronr3); % Filtrage
DSP_x3 = pwelch(signalfiltre3, [], [], Fe, "twosided"); % Densité
 spectrale de puissance
DSP_x3 = DSP_x3./max(DSP_x3); % Normaliser la DSP
DSP_x3_centered = pwelch(signalfiltre3, [], [], Fe, "centered"); %
 Densité spectrale de puissance centrée
DSP_x3_centered = DSP_x3_centered./max(DSP_x3_centered); % Normaliser
 la DSP centrée
DSP_x3_dB = 10*log10(pwelch(signalfiltre3)); % Densité spectrale de
 puissance en dB

% Etude théorique
Tf3 = abs(fft(h3, Fe));
DSP_x3_theorique = (1/Ts1) * Tf3.^2; % Densité spectrale de puissance
 en théorique
```

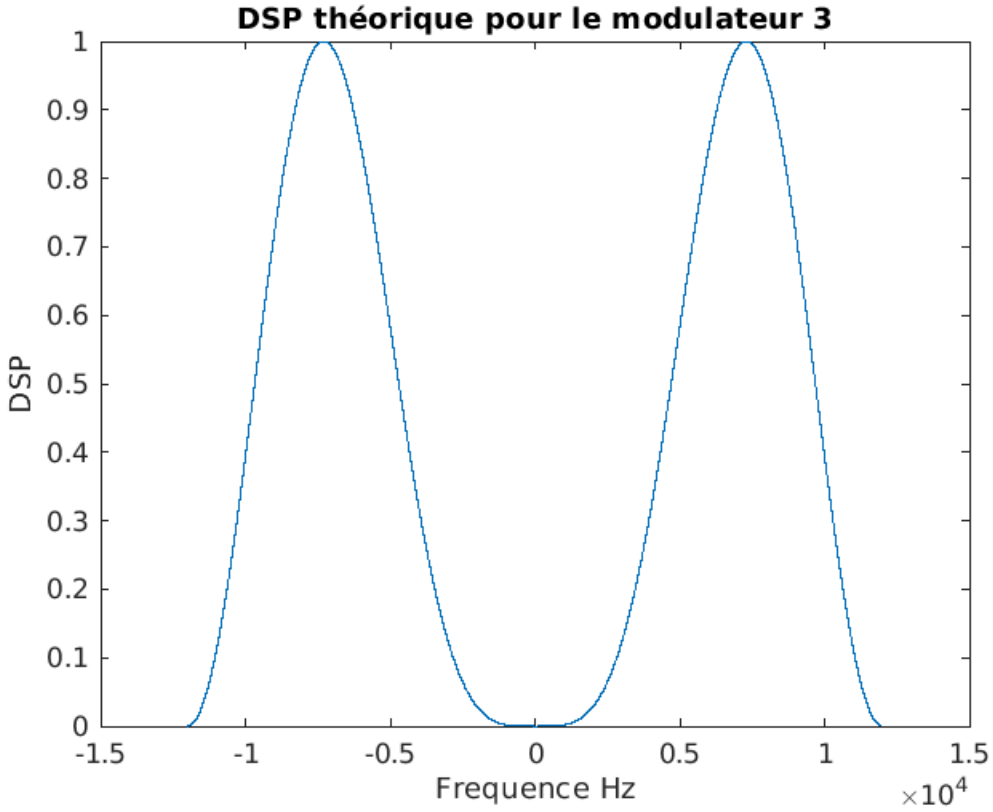
```
DSP_x3_theorique = DSP_x3_theorique./max(DSP_x3_theorique); %
 Normaliser la DSP

% Tracés
t3 = 0:Te:(length(Kronr3)-1)*Te; % Interval de temps discret
figure,plot(t3,signalfiltre3);
axis([0 0.005 -1.5 1.5]);
xlabel("Temps");
ylabel("Signal");
title("Signal du modulateur 3");

frequence3 = -Fe/2:Fe/(length(DSP_x3)-1):Fe/2; % Interval de fréquence
 discret
figure,plot(DSP_x3_dB);
ylabel("DSP (dB)");
title("DSP par estimation pour le modulateur 3 en dB");
figure,plot(frequence3, fftshift(DSP_x3_theorique));
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP théorique pour le modulateur 3");
figure,plot(frequence3, DSP_x3_centered);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP centered pour le modulateur 3");
figure,plot(frequence3, DSP_x3_theorique);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP théorique pour le modulateur 3");
figure,plot(frequence3, DSP_x3);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP par estimation twosided pour le modulateur 3");

% Comparaison DSPs Modulateur 3
figure();
title("Comparaison DSP Modulateur 3");
hold on
plot(frequence3, DSP_x3_centered, "DisplayName", "DSP-3-pratique");
plot(frequence3, fftshift(DSP_x3_theorique), "DisplayName", "DSP-3-
theorique");
hold off
legend;
```

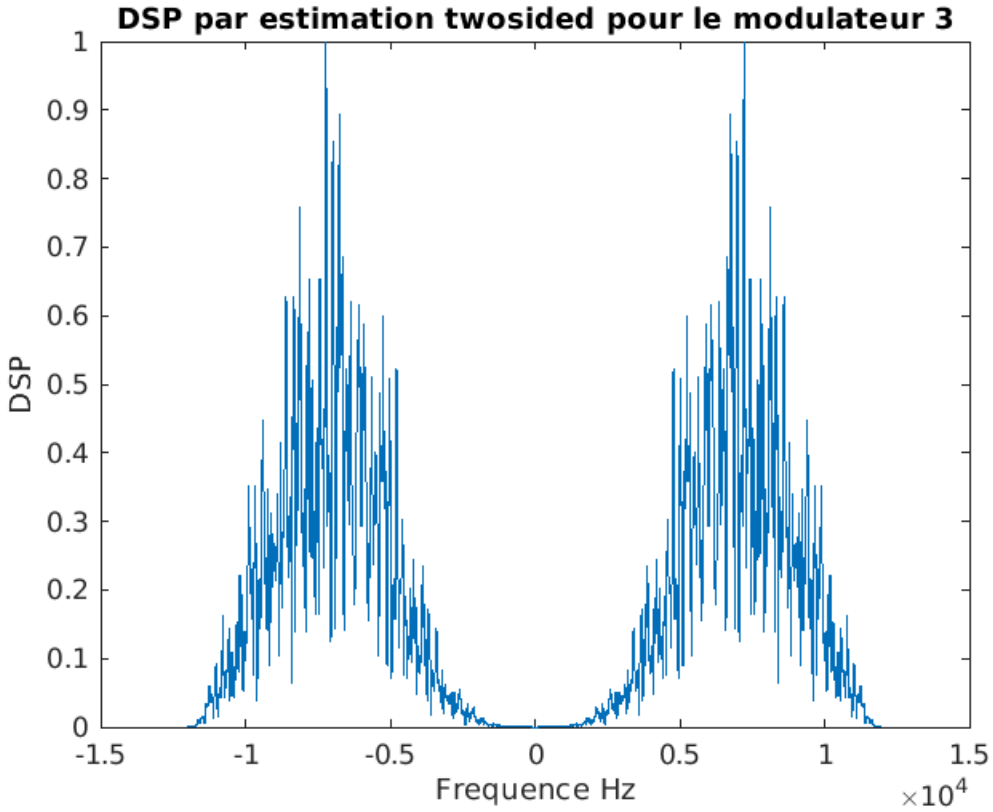

Signal du modulateur 3
Signal
Temps
×10^-3
1.5
1
0.5
0
-0.5
-1
-1.5
0
1
2
3
4
5

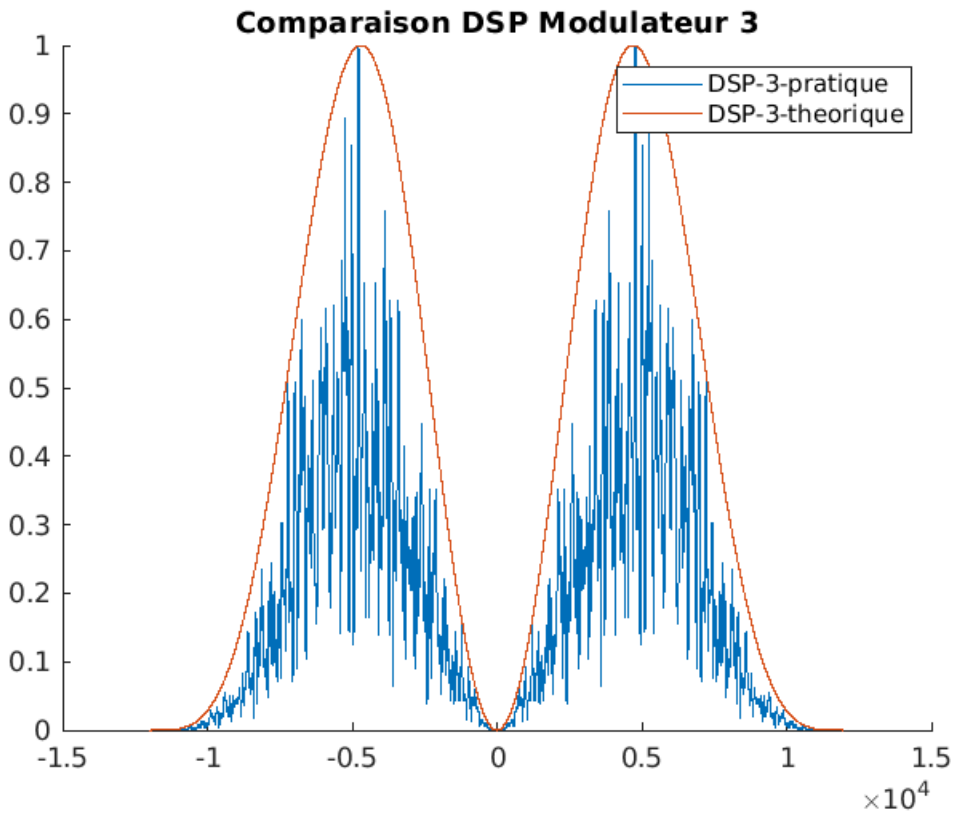

DSP par estimation pour le modulateur 3 en dB
DSP (dB)
20
0
-20
-40
-60
-80
-100
-120
0
100
200
300
400
500
600


DSP théorique pour le modulateur 3
DSP
Frequence Hz
×10^4


DSP centered pour le modulateur 3
DSP
Frequence Hz
×10^4


DSP théorique pour le modulateur 3
DSP
Frequence Hz
×10^4


DSP par estimation twosided pour le modulateur 3
DSP
Frequence Hz
×10^4

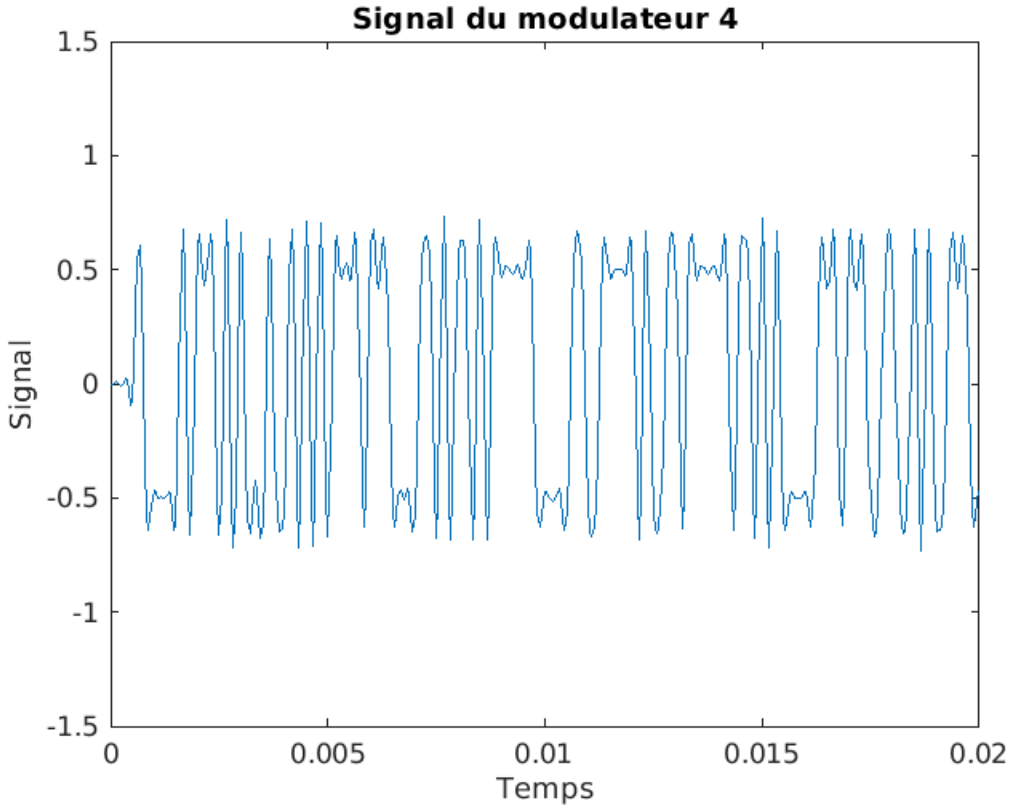


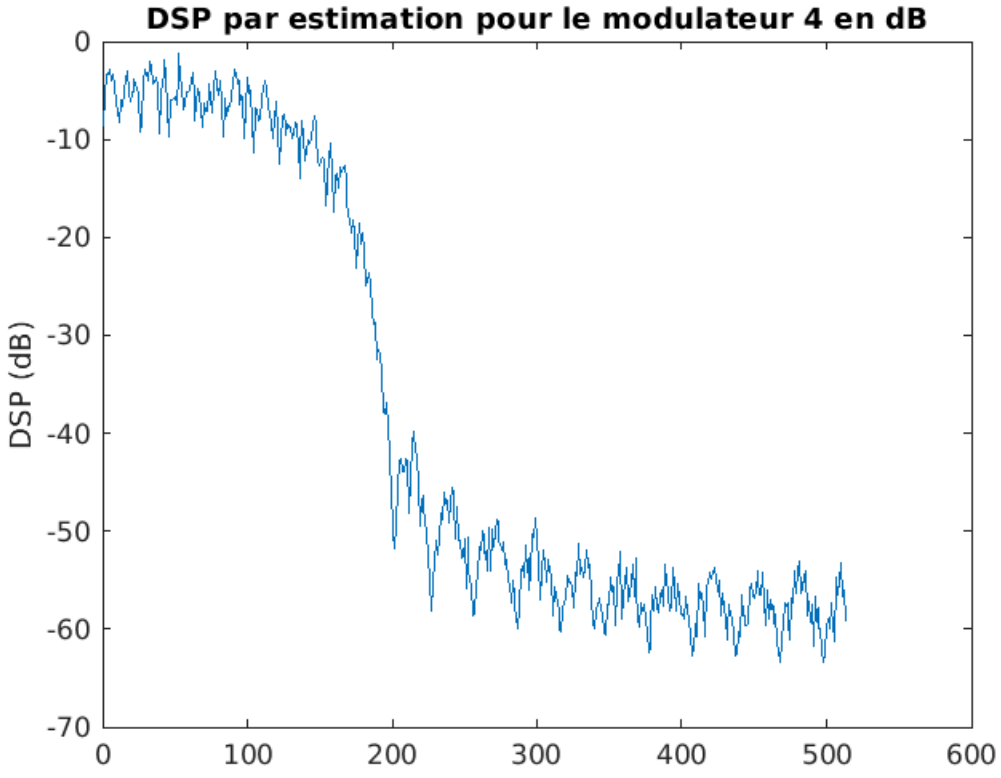
Modulateur 4

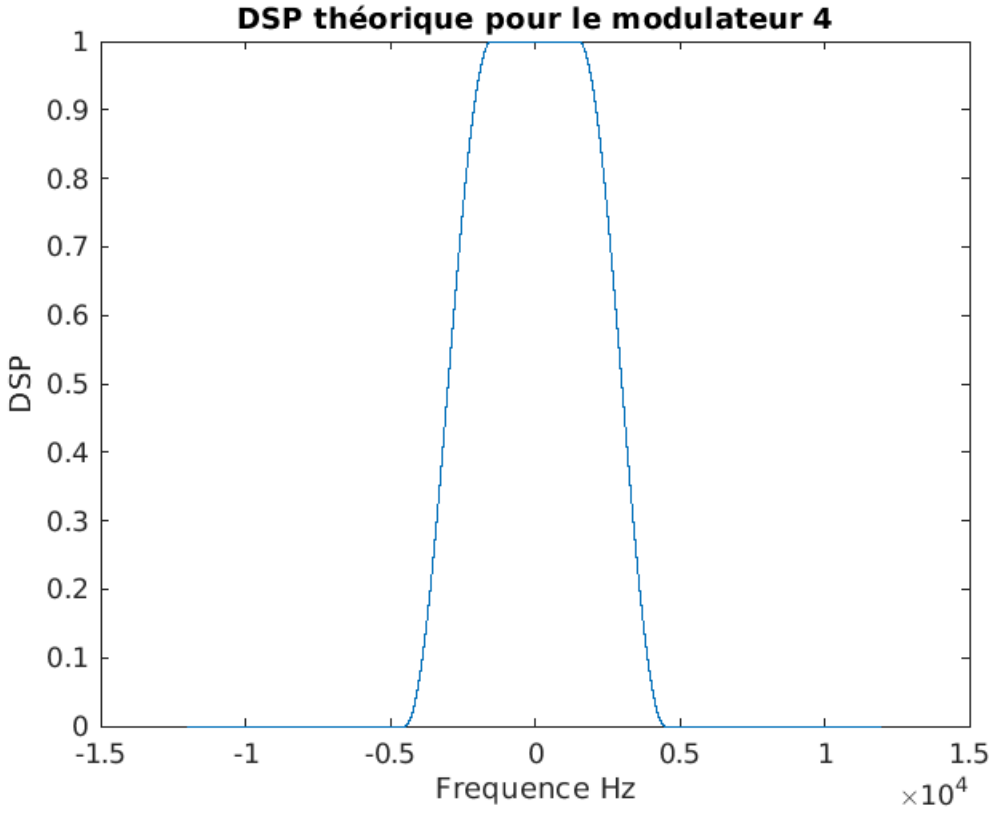
Mapping

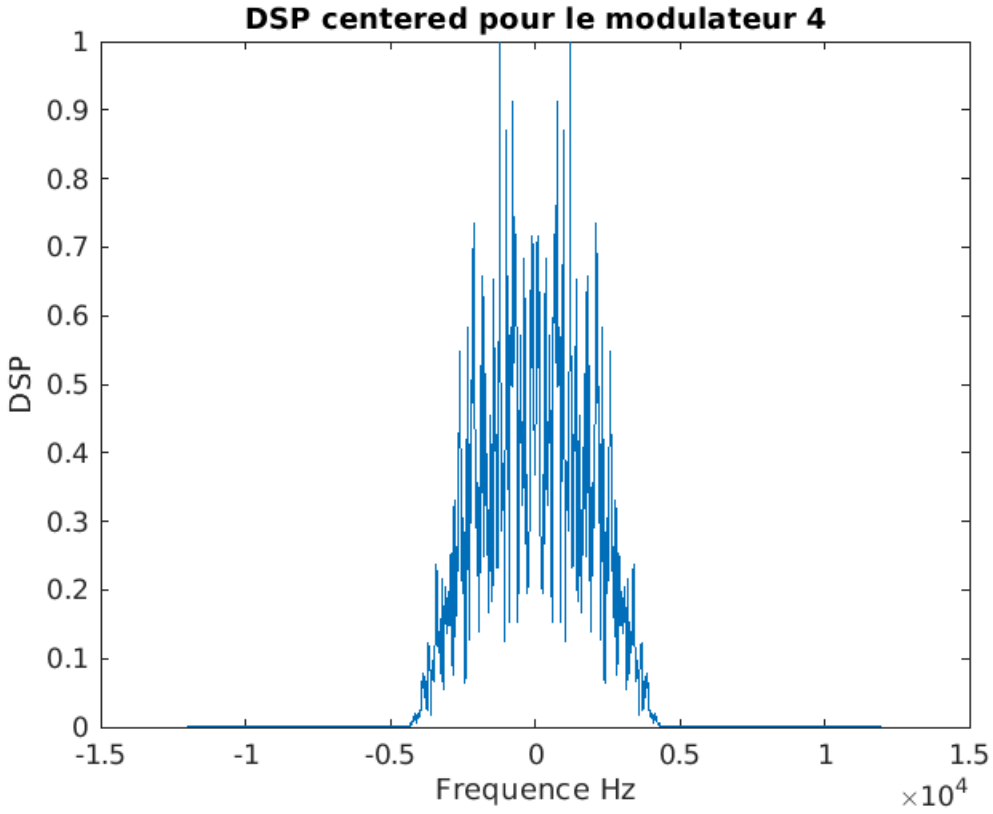
```
mapp4 = Donnees - (Donnees == 0);
% Suréchantillonage
h4 = rcosdesign(0.5,8,Ns1); % Filtre en cos

% Étude par estimation
Kronr4 = kron(mapp4, [1 zeros(1, Ns1-1)]); % Positionnement des zéros
 entre les ak
signalfiltre4=filter(h4, 1, Kronr4); % Filtrage
DSP_x4 = pwelch(signalfiltre4, [], [], Fe, "twosided"); % Densité
 spectrale de puissance
DSP_x4 = DSP_x4./max(DSP_x4); % Normaliser la DSP
DSP_x4_centered = pwelch(signalfiltre4, [], [], Fe, "centered"); %
 Densité spectrale de puissance centrée
DSP_x4_centered = DSP_x4_centered./max(DSP_x4_centered); % Normaliser
 la DSP centrée
DSP_x4_dB = 10*log10(pwelch(signalfiltre4));% Densité spectrale de
 puissance en dB

% Étude théorique
% Calcul du sigma a
mapp4_centres = abs(mapp4 - repmat(mean(mapp4), 1, length(mapp4)));
sigma_a = mean(mapp4_centres.^2);
```

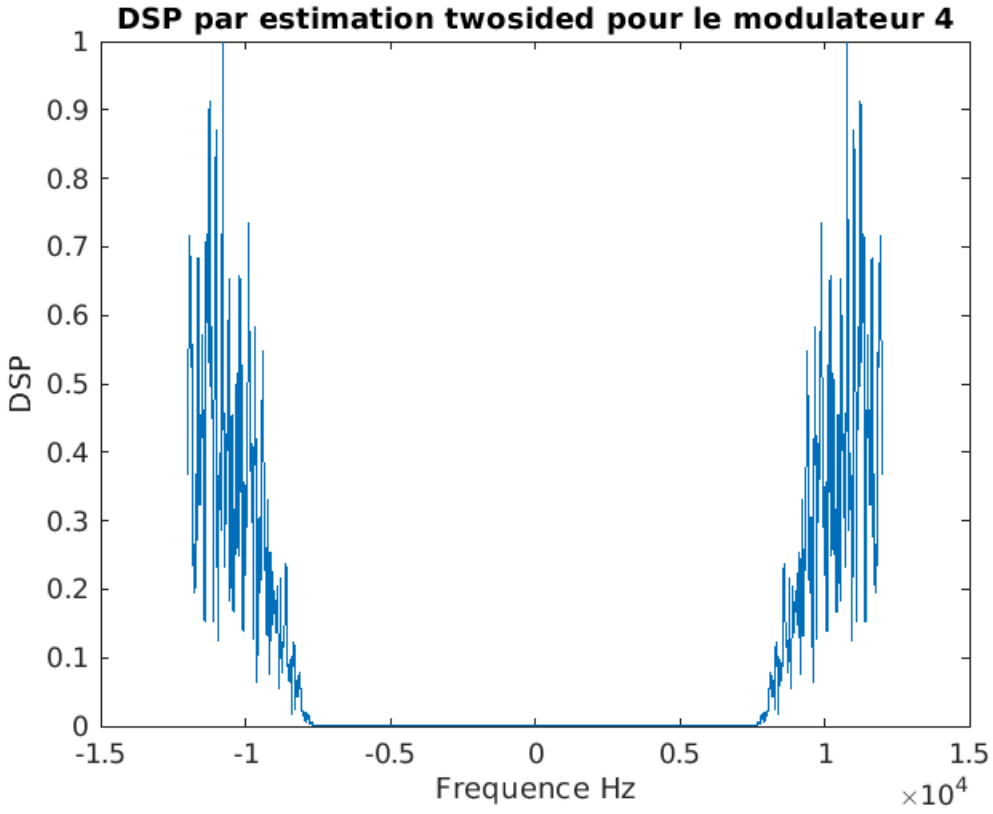
```
% Calcul de la DSP
frequence4th = -Fe/2:Fe/(length(DSP_x4)-1):Fe/2;
DSP_x4_theorique = zeros(1, length(frequence4th)); % Initialiser la
 DSP à 0
alpha = 0.5;
Condition1 = abs(frequence4th) <= (1-alpha)/(2*Ts1);
Condition2 = ((abs(frequence4th)<=(1+alpha)/(2*Ts1)) &
 (abs(frequence4th)>(1-alpha)/(2*Ts1)));
i = find(Condition2 == 1);
DSP_x4_theorique(Condition1) = sigma_a; % DSP lorsque |f| <= (1-
alpha)/2Ts
DSP_x4_theorique(Condition2 == 1) = (sigma_a/2) .* ( ones(1,
 length(i)) + cos((pi*Ts1/alpha).*(abs(frequence4th(i)) - ((1-alpha)/
(2*Ts1)).*ones(1,length(i)))));
DSP_x4_theorique = DSP_x4_theorique./max(DSP_x4_theorique); %
 Normaliser la DSP

% Tracés
t4 = 0:Te:(length(Kronr4)-1)*Te; % Interval de temps discret
figure,plot(t4,signalfiltre4);
axis([0 0.02 -1.5 1.5]);
xlabel("Temps");
ylabel("Signal");
title("Signal du modulateur 4");

frequence4 = -Fe/2:Fe/(length(DSP_x4)-1):Fe/2; % Interval de fréquence
 discret
figure,plot(DSP_x4_dB);
ylabel("DSP (dB)");
title("DSP par estimation pour le modulateur 4 en dB");
figure,plot(frequence4th, DSP_x4_theorique);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP théorique pour le modulateur 4");
figure,plot(frequence4, DSP_x4_centered);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP centered pour le modulateur 4");
figure,plot(frequence4, DSP_x4);
xlabel("Frequence Hz");
ylabel("DSP");
title("DSP par estimation twosided pour le modulateur 4");

% Comparaison DSPs Modulateur 4
figure();
title("Comparaison DSP Modulateur 4");
hold on
plot(frequence4, DSP_x4_centered, "DisplayName", "DSP-4-pratique");
plot(frequence4th, DSP_x4_theorique, "DisplayName", "DSP-4-
theorique");
hold off
legend;
```

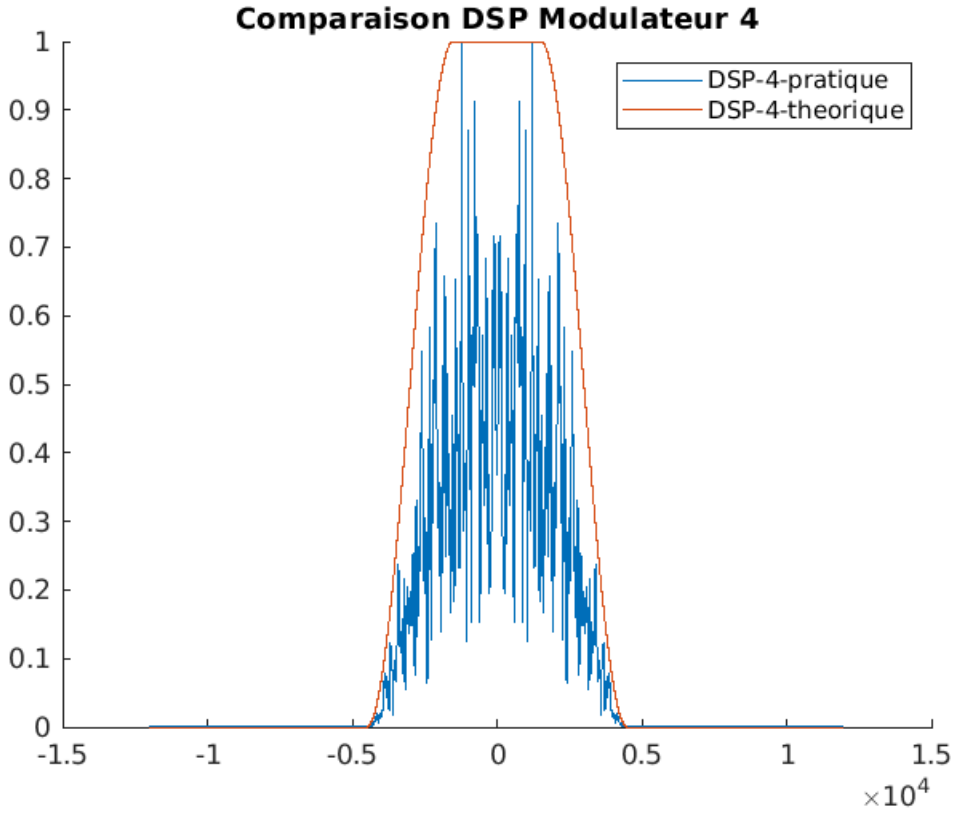

Signal du modulateur 4
Signal
Temps
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.005
0.01
0.015
0.02


DSP par estimation pour le modulateur 4 en dB
DSP (dB)
0
-10
-20
-30
-40
-50
-60
-70
0
100
200
300
400
500
600


DSP théorique pour le modulateur 4
DSP
Frequence Hz
×10^4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-1.5
-1
-0.5
0
0.5
1
1.5


DSP centered pour le modulateur 4
DSP
Frequence Hz
×10^4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-1.5
-1
-0.5
0
0.5
1
1.5

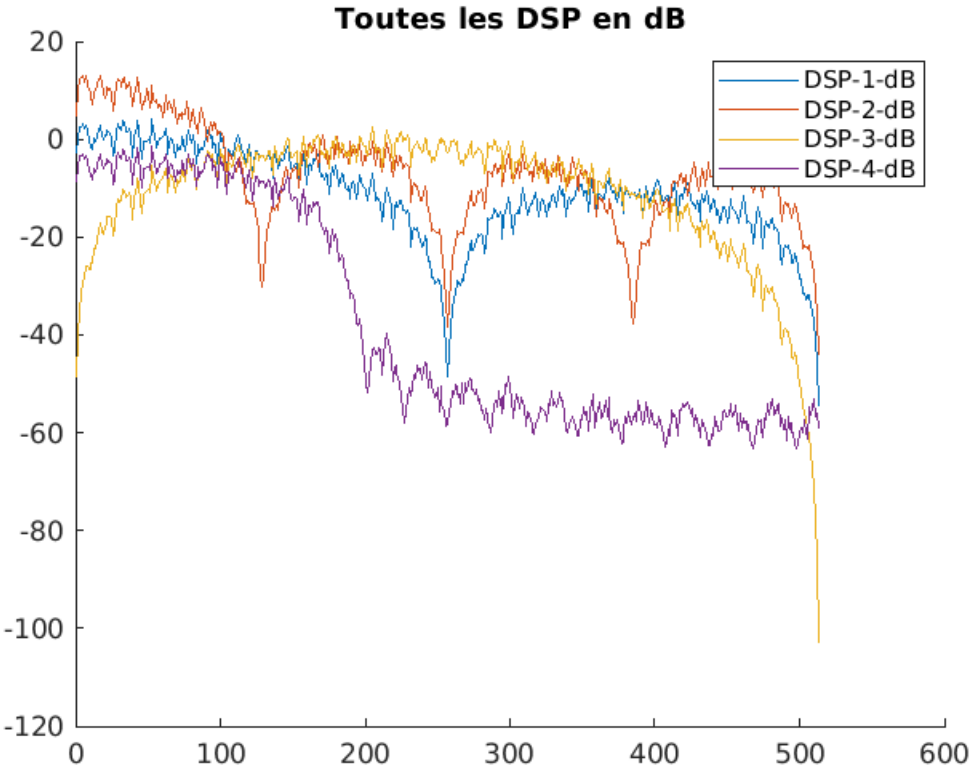

DSP par estimation twosided pour le modulateur 4
DSP
Frequence Hz
×10^4
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1.5
-1
-0.5
0
0.5
1
1.5

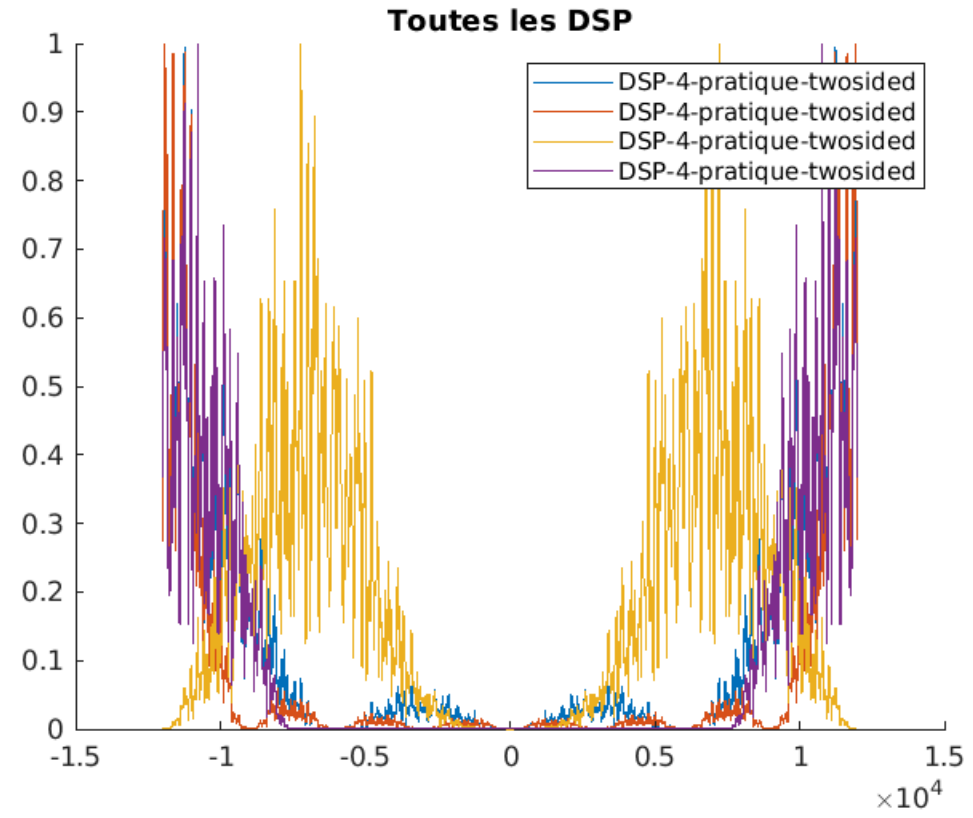

Comparaison DSP Modulateur 4
DSP-4-pratique
DSP-4-theorique
×10^4
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1.5
-1
-0.5
0
0.5
1
1.5

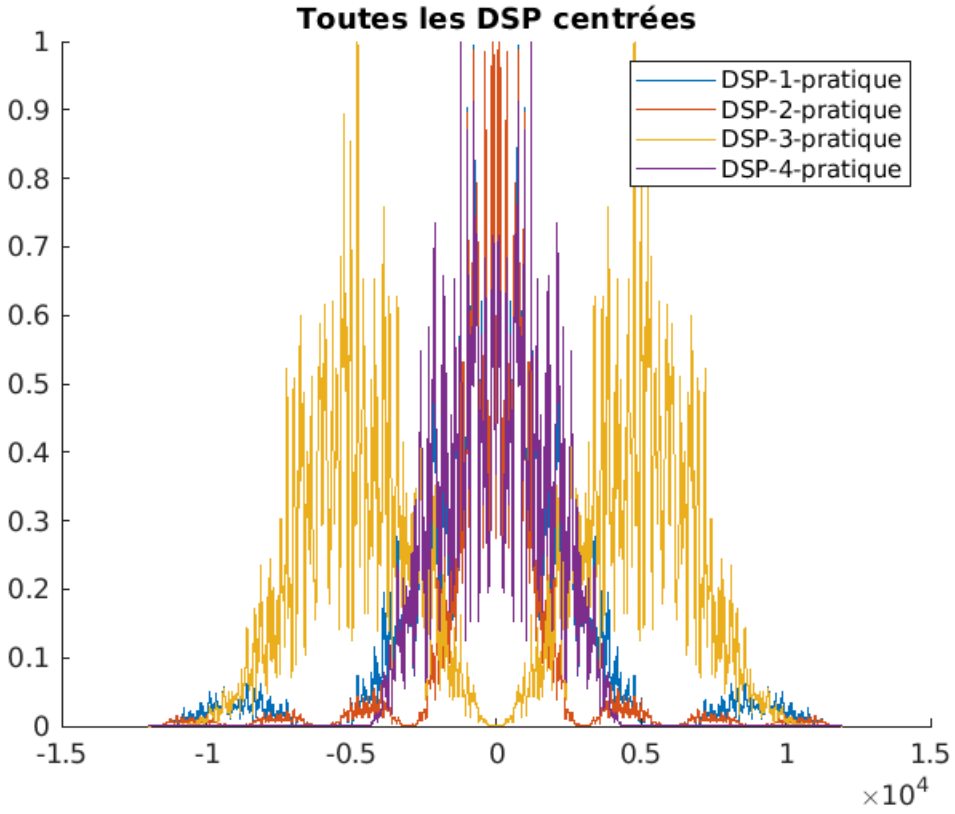
Comparaison des DSP

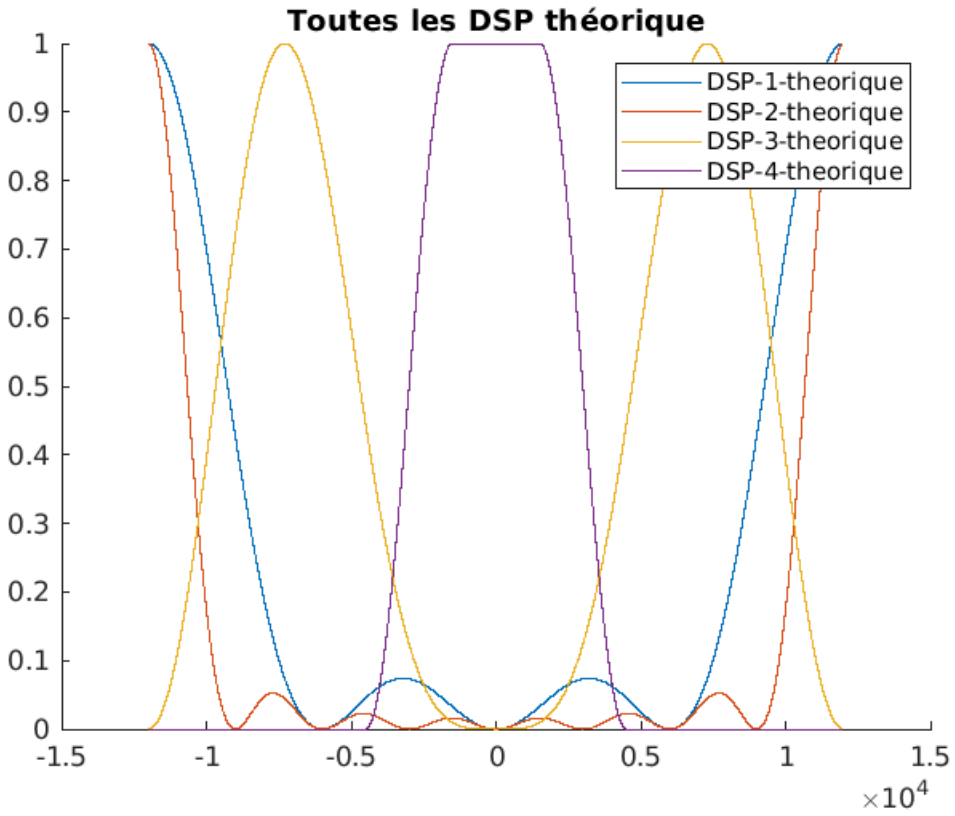
DSP en dB

```
figure();
title("Toutes les DSP en dB");
hold on
plot(DSP_x1_dB, "DisplayName", "DSP-1-dB");
plot(DSP_x2_dB, "DisplayName", "DSP-2-dB");
plot(DSP_x3_dB, "DisplayName", "DSP-3-dB");
plot(DSP_x4_dB, "DisplayName", "DSP-4-dB");
hold off
legend;

% Toutes les DSP
figure();
title("Toutes les DSP");
hold on
plot(frequence1, DSP_x1, "DisplayName", "DSP-4-pratique-twosided");
plot(frequence2, DSP_x2, "DisplayName", "DSP-4-pratique-twosided");
plot(frequence3, DSP_x3, "DisplayName", "DSP-4-pratique-twosided");
plot(frequence4, DSP_x4, "DisplayName", "DSP-4-pratique-twosided");
hold off
legend;

% Toutes les DSP centrées
figure();
title("Toutes les DSP centrées");
hold on
plot(frequence1, DSP_x1_centered, "DisplayName", "DSP-1-pratique");
plot(frequence2, DSP_x2_centered, "DisplayName", "DSP-2-pratique");
plot(frequence3, DSP_x3_centered, "DisplayName", "DSP-3-pratique");
plot(frequence4, DSP_x4_centered, "DisplayName", "DSP-4-pratique");
hold off
legend;

% Toutes les DSP théorique
figure();
title("Toutes les DSP théorique");
hold on
plot(frequence1, DSP_x1_theorique, "DisplayName", "DSP-1-theorique");
plot(frequence2, DSP_x2_theorique, "DisplayName", "DSP-2-theorique");
plot(frequence3, DSP_x3_theorique, "DisplayName", "DSP-3-theorique");
plot(frequence4, DSP_x4_theorique, "DisplayName", "DSP-4-theorique");
hold off
legend;
```


Toutes les DSP en dB
DSP-1-dB
DSP-2-dB
DSP-3-dB
DSP-4-dB
20
0
-20
-40
-60
-80
-100
-120
0
100
200
300
400
500
600


Toutes les DSP
DSP-4-pratique-twosided
DSP-4-pratique-twosided
DSP-4-pratique-twosided
DSP-4-pratique-twosided
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1.5
-1
-0.5
0
0.5
1
1.5
×10^4


Toutes les DSP centrées
DSP-1-pratique
DSP-2-pratique
DSP-3-pratique
DSP-4-pratique
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1.5
-1
-0.5
0
0.5
1
1.5
×10^4


Toutes les DSP théorique
DSP-1-theorique
DSP-2-theorique
DSP-3-theorique
DSP-4-theorique
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1.5
-1
-0.5
0
0.5
1
1.5
×10^4

Bandes

Bande et efficacité pour le premier modulateur (x% = 95%)

```
find_1 = find(fftshift(DSP_x1_theorique) >= 0.05);
Bande_1 = (max(find_1) - min(find_1))/2;
efficace_1 = Rb/Bande_1;
% Bande et efficacité pour le deuxième modulateur
find_2 = find(fftshift(DSP_x2_theorique) >= 0.05);
Bande_2 = (max(find_2) - min(find_2))/2;
efficace_2 = Rb/Bande_2;
% Bande et efficacité pour le troisième modulateur
find_3 = find(fftshift(DSP_x3_theorique) >= 0.05);
Bande_3 = (max(find_3) - min(find_3))/2;
efficace_3 = Rb/Bande_3;
% Bande et efficacité pour le quatrième modulateur
find_4 = find(DSP_x4_theorique >= 0.05);
Bande_4 = (max(find_4) - min(find_4))/2;
efficace_4 = Rb/Bande_4;
Liste = [efficace_1 efficace_2 efficace_3 efficace_4];
```

Classement

```
C = {'Modulateur 1 d´efficacité égale à','Modulateur 2 d´efficacité
 égale à','Modulateur 3 d´efficacité égale à','Modulateur 4 d
´efficacité égale à'};
[new,idx] = sort(Liste);
D = C(idx); % Trier les modulateurs
D(2,:) = num2cell(new);
% Le classement est
fprintf('%10s %d\n',D{:})
```

```
Modulateur 1 d´efficacité égale à 5.995803e-01
Modulateur 3 d´efficacité égale à 6.205399e-01
Modulateur 2 d´efficacité égale à 1.324796e+00
Modulateur 4 d´efficacité égale à 1.474745e+00
```

Published with MATLAB® R2020a