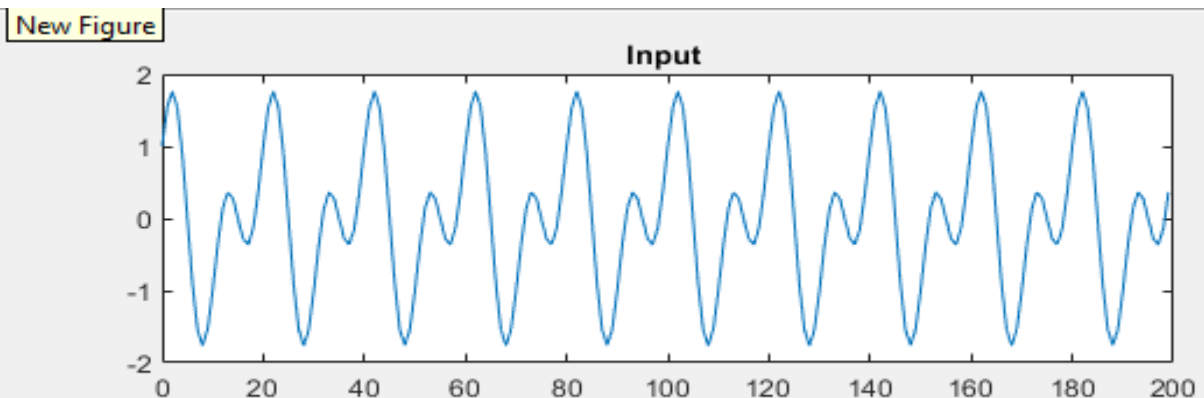


Contenus :

- Début
- Création du signal
- Filtre 1
- Sortie 1
- Filtre 2
- Sortie 2
- Comparer les réponses d'amplitude
- Compare les phases de réponses
- Ajout du bruit
- Filtrage du signal bruyant
- Réponses impulsionnels
- Conclusion

Création du signal:

```
N = 200;  
n = 0:N-1;  
  
om1 = 0.1*pi;  
om2 = 0.2*pi;  
  
x = cos(om1*n) + sin(om2*n);      % Input signal  
  
figure(1)  
clf  
subplot(2, 1, 1)  
plot(n, x)  
title('Input ')
```



Filtre 1 :

```

%% Filter 1

load filter1.mat % loads b1, a1 % Load filter 1 (computed previously and saved)
b1 % b1, a1 : coefficients of difference equation
a1

[H1, om] = freqz(b1, a1); % Compute frequency response
f = om/(2*pi); % Frequency (cycles/sample)

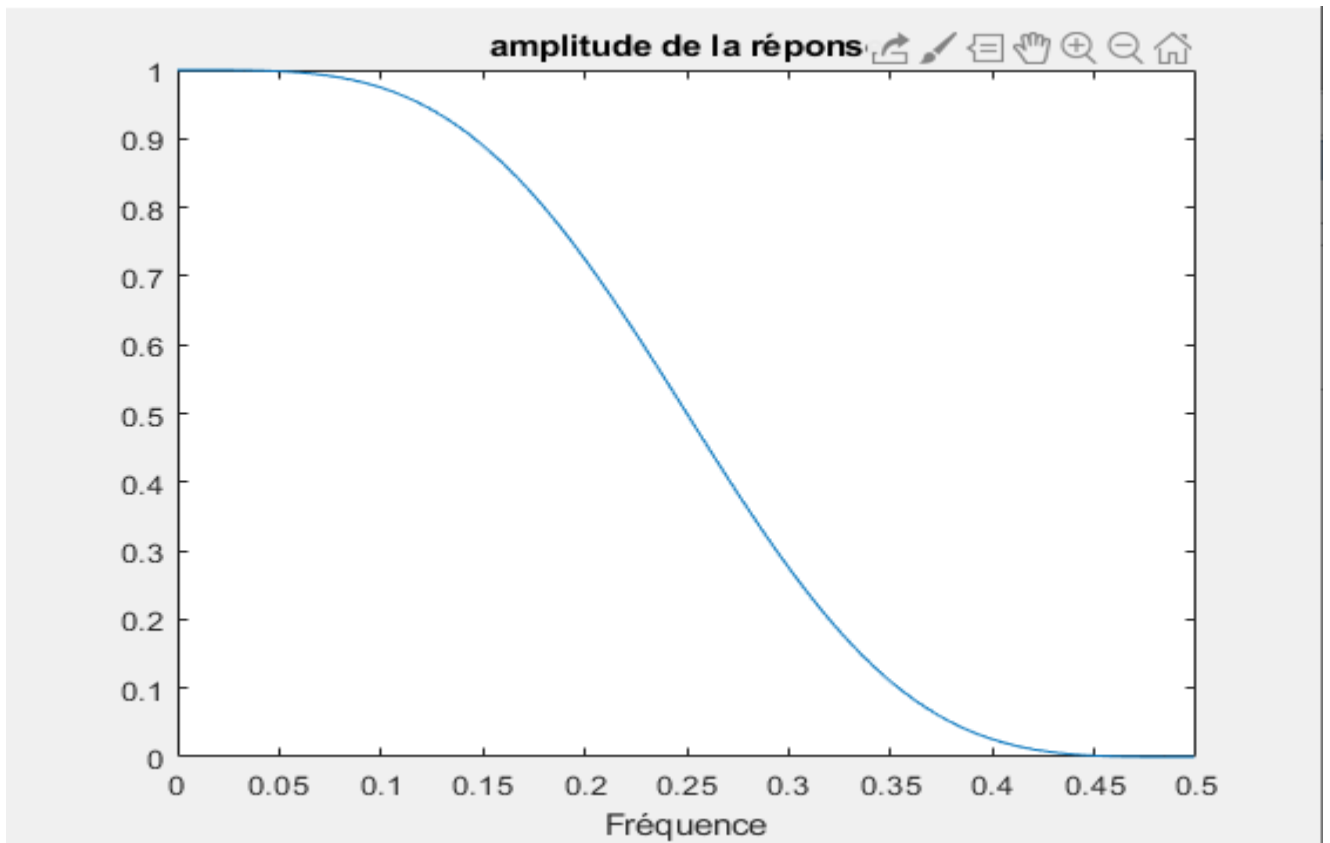
figure(1)
clf
plot(f, abs(H1))
xlabel('Fréquence')
title('amplitude de la réponse')

```

```

b1 = -0.0312    0  0.2812  0.5000  0.2812    0 -0.0312
a1 = 1

```



Sortie 1 :

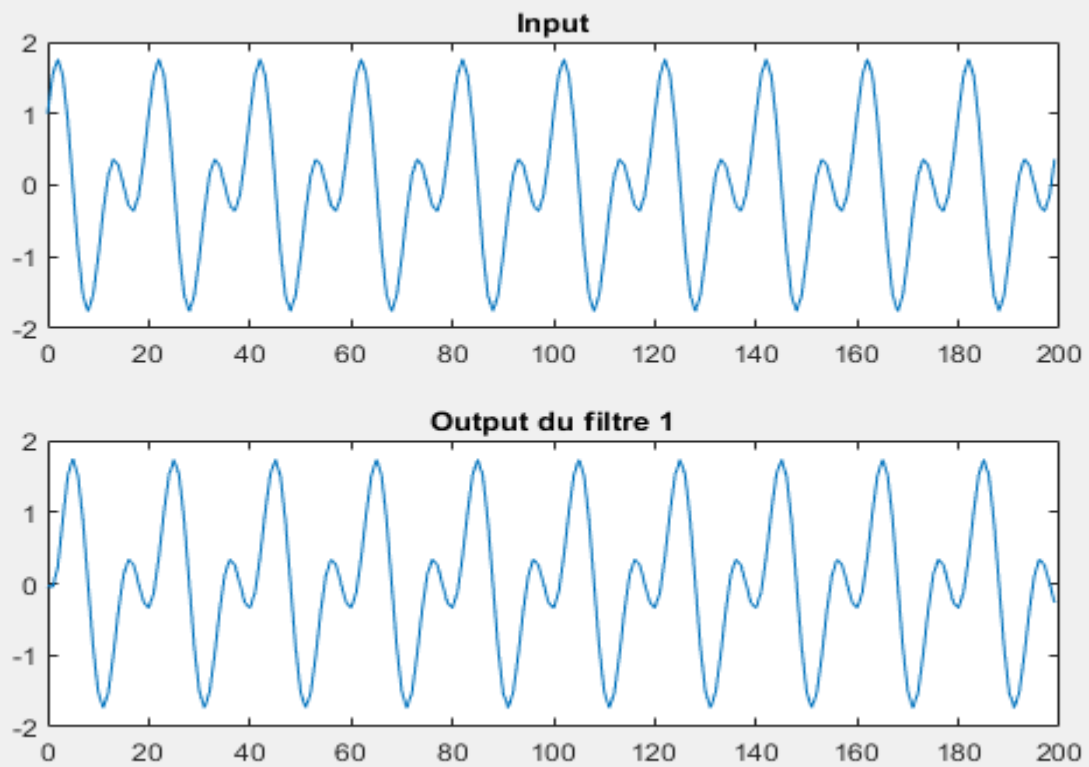
Le signal de sortie ressemble au signal d'entrée, sauf retardé. Le signal d'entrée est dans la bande passante du filtre.

```

y1 = filter(b1, a1, x);           % Apply filter 1 to signal x

figure(1)
clf
subplot(2, 1, 1)
plot(n, x)
title('Input')
subplot(2, 1, 2)
plot(n, y1)
title('Output du filtre 1')

```



Filtre 2 :

```

%% Filter 2

load filter2.mat % loads b2 a2      % Load filter 2 (previously computed and saved)
b2
a2

[H2, om] = freqz(b2, a2);          % Compute frequency response

figure(1)
clf
plot(f, abs(H2))
title('amplitude de la réponse')
xlabel('Fréquence')

```

b2 =

Colonnes 1 à 7

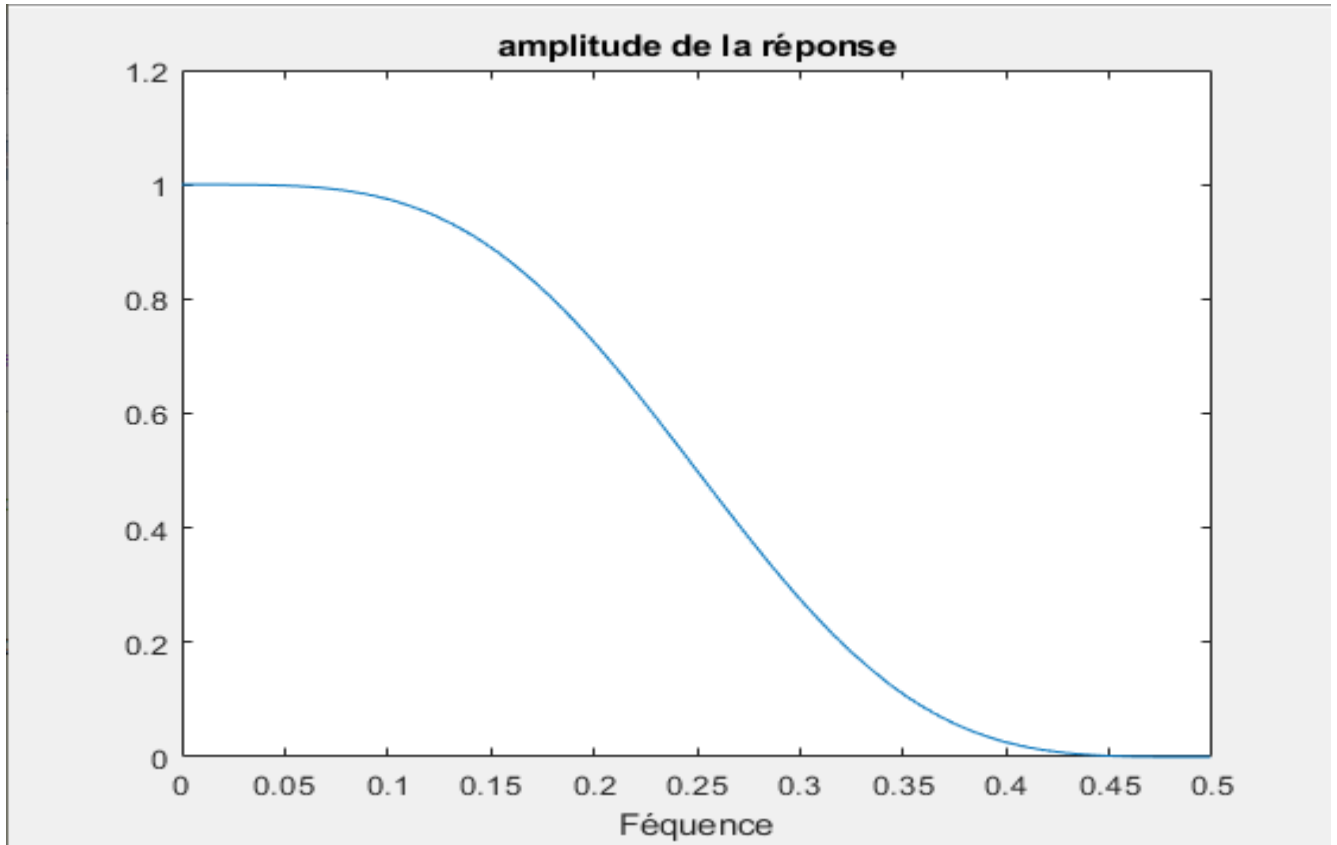
-0.0253 0.0523 0.1966 -0.0657 -0.3277 0.0293 0.2559

Colonnes 8 à 9

0.0523 -0.0312

a2 =

1.0000 -1.6736 0.8100

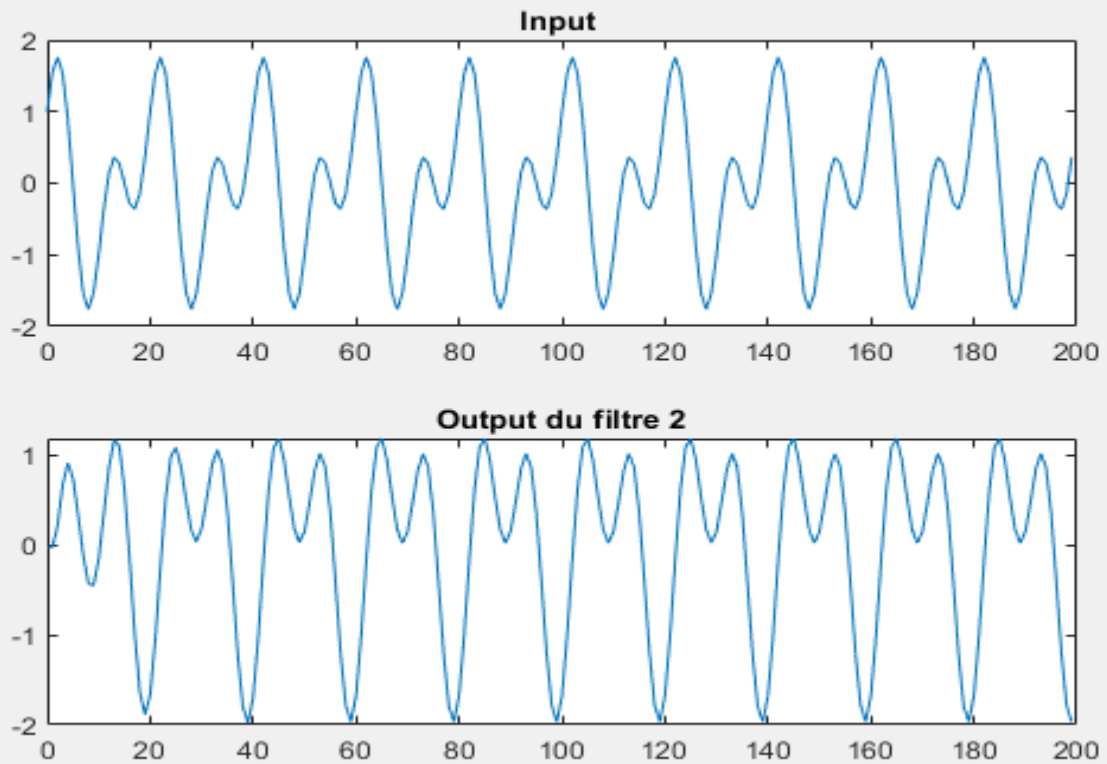


Sortie 2 :

La sortie 2 diffère en forme d'onde de la sortie 1. Pourquoi ?

```
y2 = filter(b2, a2, x);           % Apply filter 2 to signal x

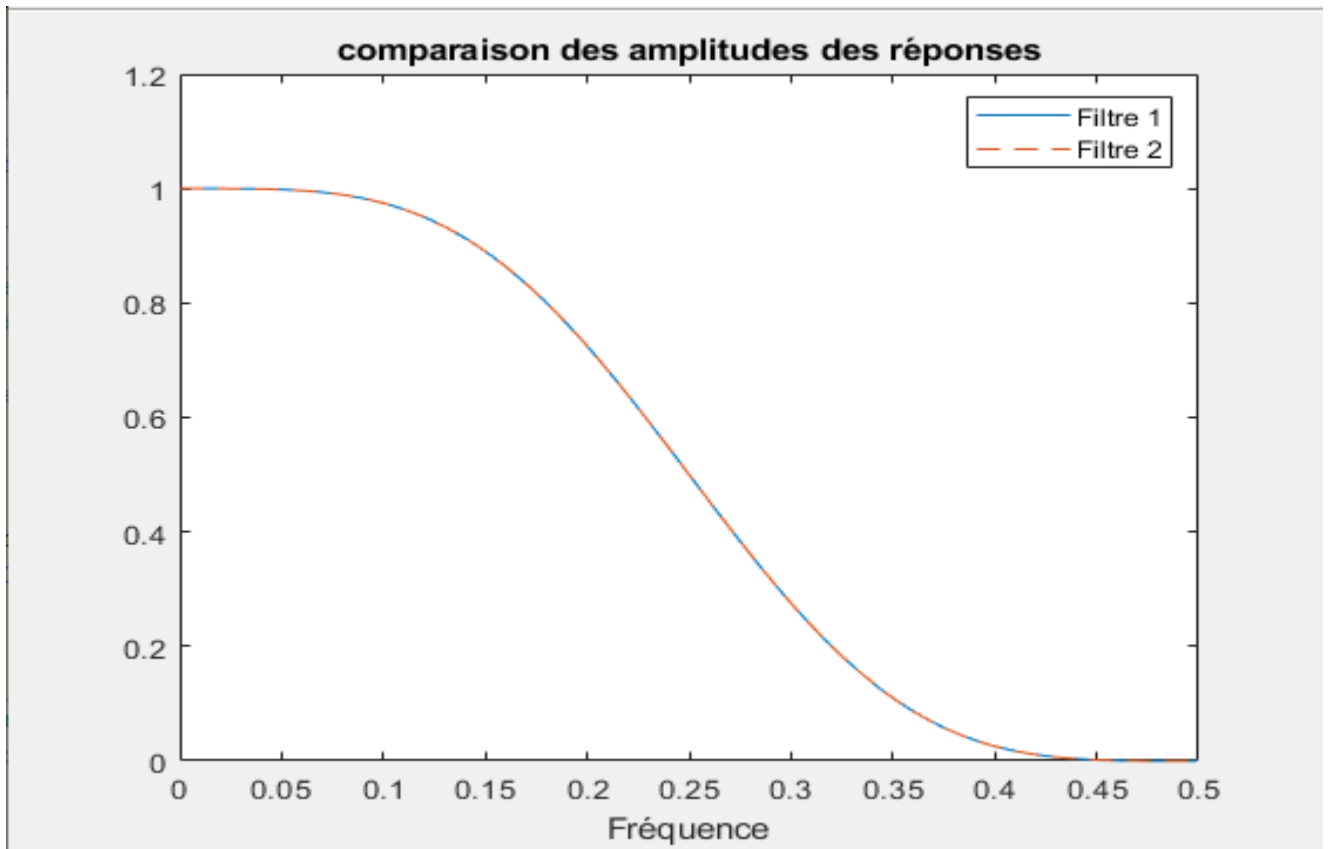
figure(1)
subplot(2, 1, 1)
plot(n, x)
title('Input ')
subplot(2, 1, 2)
plot(n, y2)
title('Output du filtre 2')
```



Comparer les réponses d'amplitude :

Y1 et y2 diffèrent-ils parce que les filtres 1 et 2 ont des amplitudes de réponse en fréquence différentes ? Non, les deux filtres ont la même amplitude de réponse en fréquence.

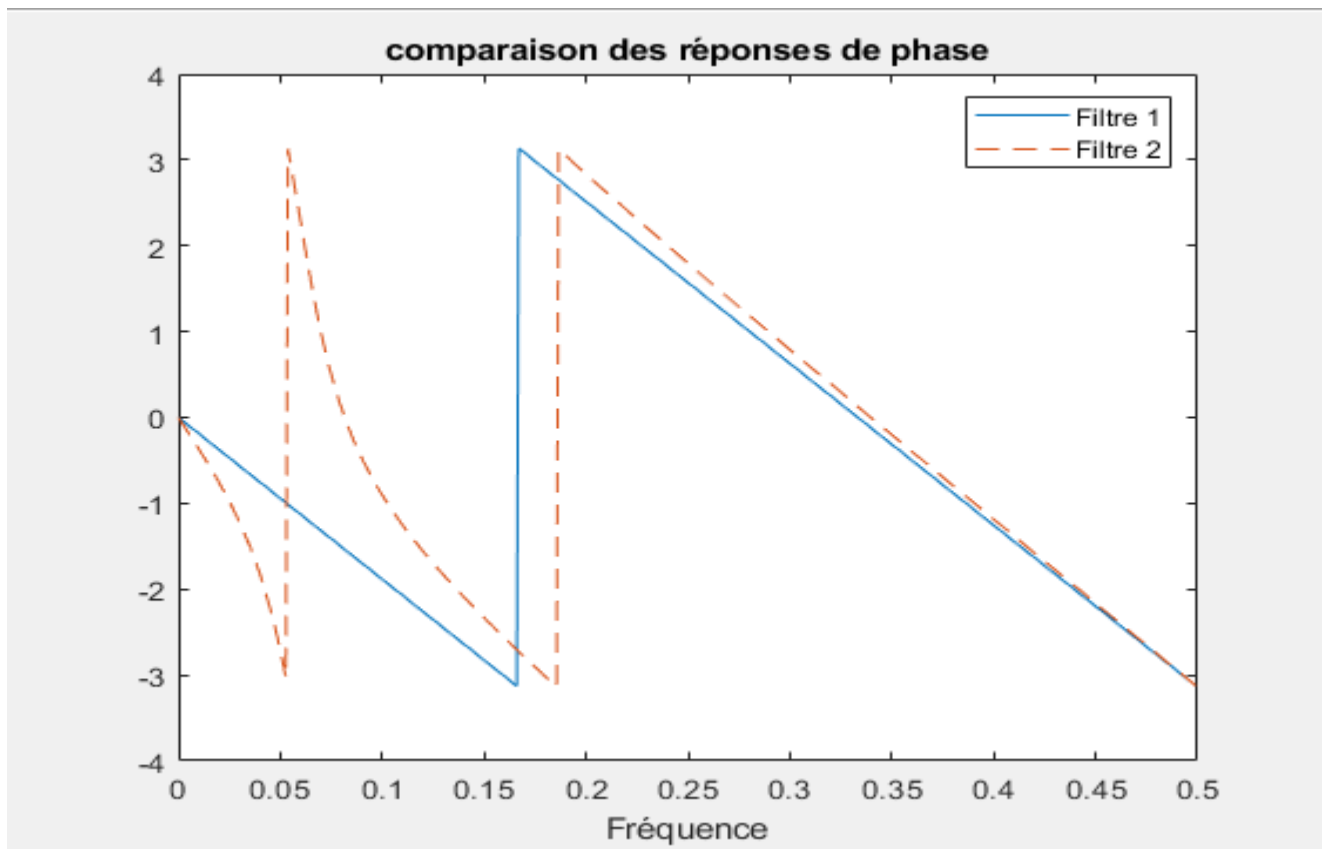
```
figure(1)
clf
plot(f, abs(H1), '-', f, abs(H2), '--')
legend('Filtre 1', 'Filtre 2')
xlabel('Fréquence')
title('comparaison des amplitudes des réponses')
```



Compare les phases de réponses

Les filtres ont-ils des réponses de phase différentes ?

```
figure(1)
clf
plot(f, angle(H1), f, angle(H2), '--')
legend('Filtre 1', 'Filtre 2')
title('comparaison des réponses de phase')
xlabel('Fréquence')
```



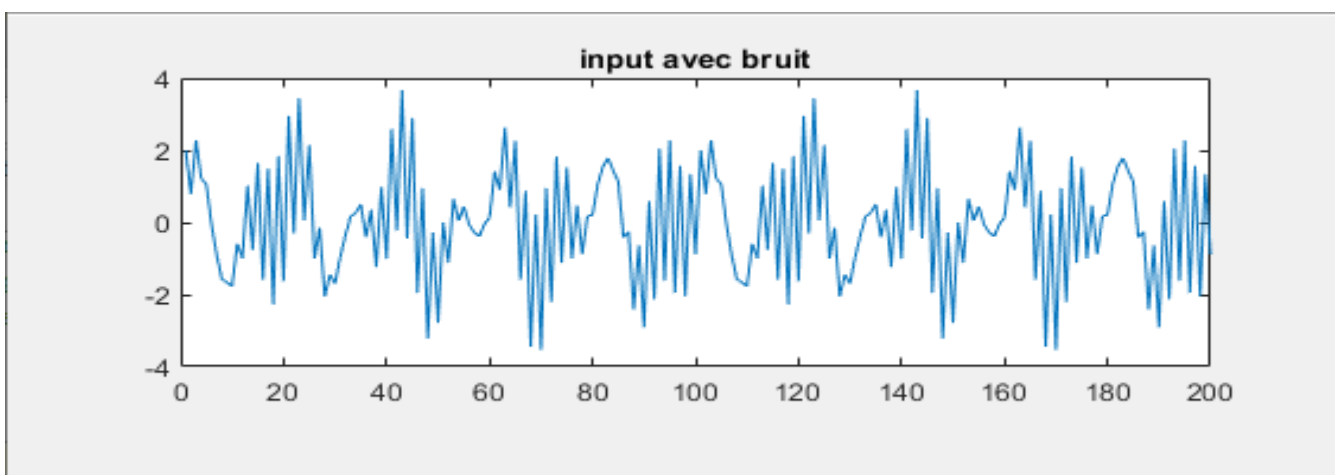
Ajout du bruit :

```
%% Add noise

load noise_data.mat           % load variable 'noise' (previously computed and saved)

r = x + noise;                % Add noise to input signal

figure(1)
clf
subplot(2, 1, 1)
plot(r)
title('input avec bruit')
```



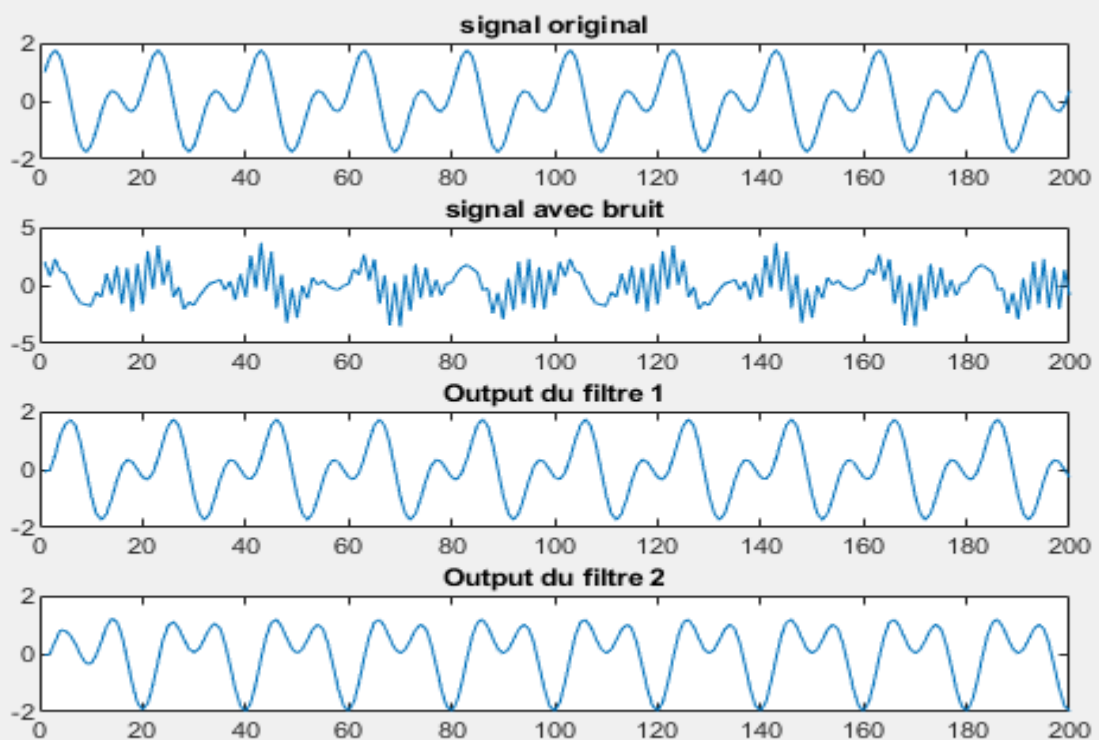
Filtrage du signal bruyant :

Le filtre 1 supprime le bruit et préserve la forme d'onde du signal. Le filtre 2 supprime le bruit mais déforme la forme d'onde du signal.

```
y1 = filter(b1, a1, r);
y2 = filter(b2, a2, r);

figure(1)
clf
subplot(4, 1, 1)
plot(x)
title('signal original')
subplot(4, 1, 2)
plot(r)
title('signal avec bruit')
subplot(4, 1, 3)
plot(y1)
title('Output du filtre 1')
subplot(4, 1, 4)
plot(y2)
title('Output du filtre 2')

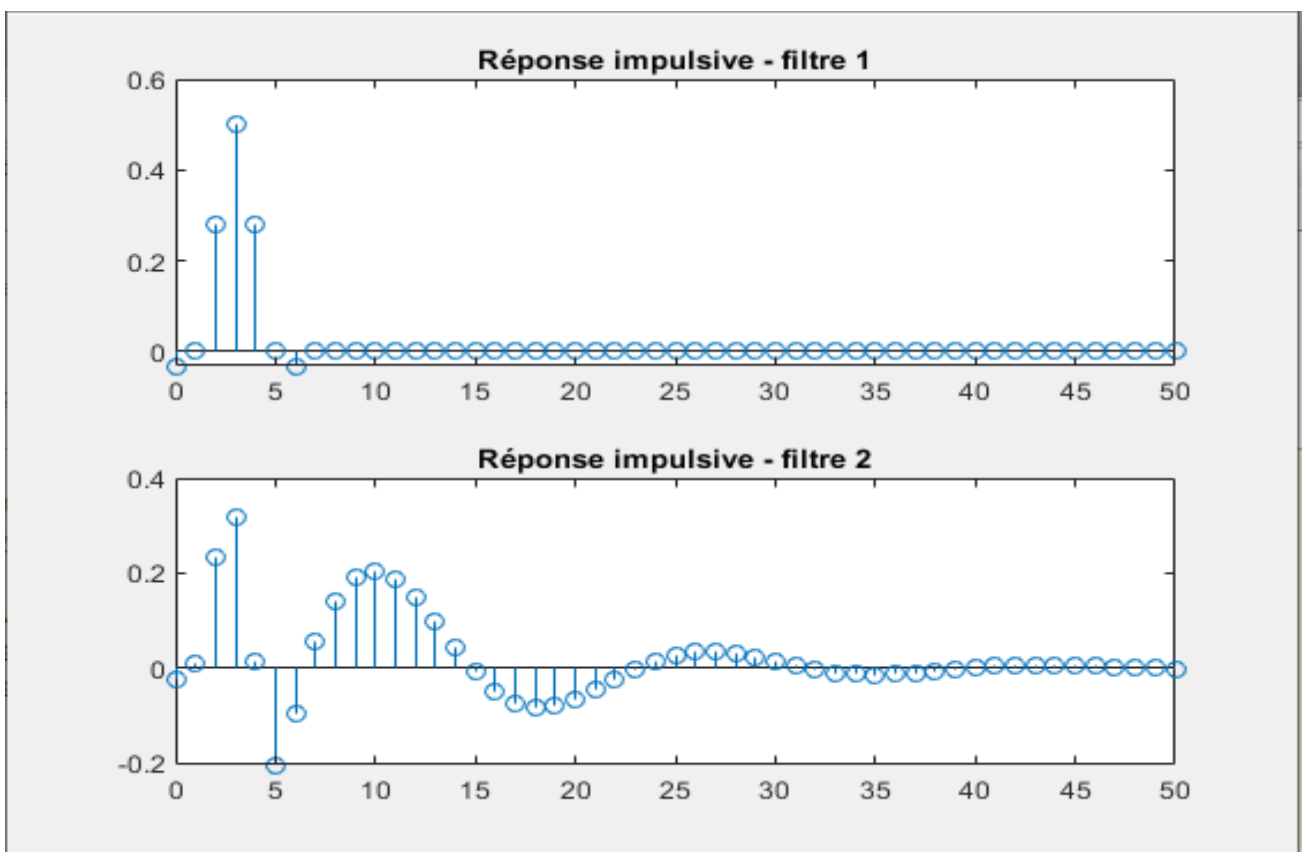
orient tall
print -dpdf compare_signals
```



Réponses impulsionnels :

Le filtre 1 a une réponse impulsionnelle symétrique (il a une phase linéaire). Le filtre 2 a une réponse impulsionnelle non symétrique (il n'a pas de phase linéaire).

```
imp = [1 zeros(1, 50)];  
h1 = filter(b1, a1, imp);  
h2 = filter(b2, a2, imp);  
  
figure(1)  
clf  
subplot(2, 1, 1)  
stem(0:50, h1)  
title('Réponse impulsive - filtre 1')  
  
subplot(2, 1, 2)  
stem(0:50, h2)  
title('Réponse impulsive - filtre 2')
```



Conclusion :

Le filtre de phase linéaire (filtre 1) est préférable car il supprime le bruit sans modifier (distorsion de) la forme d'onde.