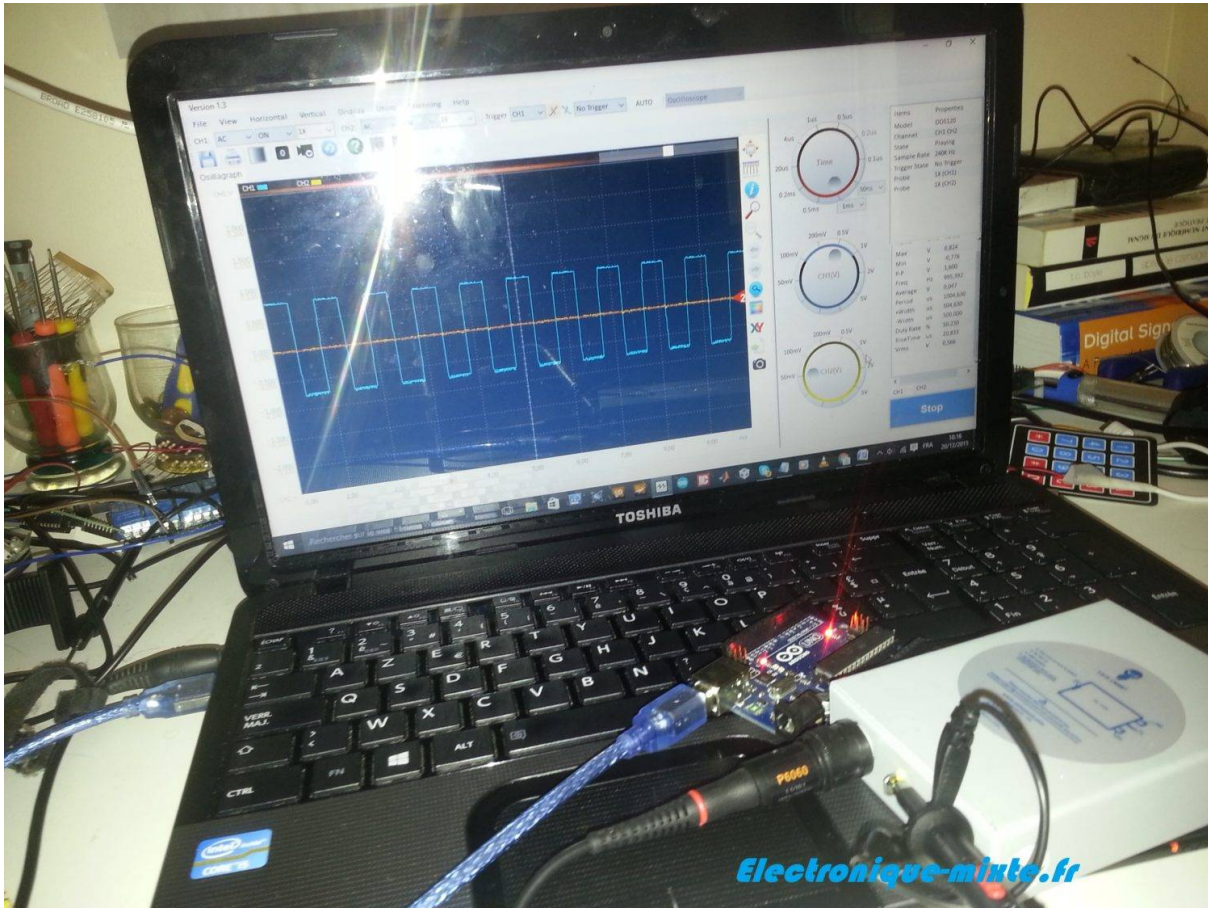




Projet électronique : Traitement du signal avec Arduino # Lissage & Seuillage d'un signal 1/3

[Total : 0 Moyenne : 0/5]



Objectifs du projet électronique

- Etude et simulation numérique du filtre [Moyenneur](#) avec [Matlab](#) [1/3]
- Etude et simulation numérique du filtre Médiane avec [Matlab](#) [1/3]
- [Etudes et simulation numérique des techniques de la détection de seuil](#) [2/3]
 - [Seuillage classique](#)
 - [Seuillage moyenné](#)
 - [Seuillage médiane](#)
- [Etude comparatives et qualitative des méthodes](#) [2/3]
- Implémentation sur [microcontrôleur](#) en utilisant le kit [Arduino](#) [3/3]



Etude et simulation numérique du filtre Moyenneur avec Matlab

Ce filtre lisseur part du principe que la valeur d'un signal est relativement similaire à son voisinage. Il fait donc en sorte que chaque valeur du signal est peut être remplacé par la moyenne pondérée de ses valeurs précédentes. Si on applique un filtre moyenneur de taille $N=10$, cela signifie qu'on additionne tous les valeurs précédentes de la valeur courante traitée puis on divise par la taille du filtre. On obtient ainsi la formule suivante du filtre.

$$s_m(t) = \frac{1}{N} \sum s(t - i), i[1..N]$$

Exemple :

On considère un morceau du signal constitue de 10 échantillons passés, la [valeur moyenne](#) sur 10 échantillons à l'instant actuel correspond à l'application de formule ci-dessus.

$$s(t)=[10.2, 10, 10.5, 10.6, 10.4, 10.8,10.12,10,11], N=10$$
$$s_m(t) = (10.2+10+10.5+10.6+10.4+10.8+10.12+10+11)/10= 9.36$$

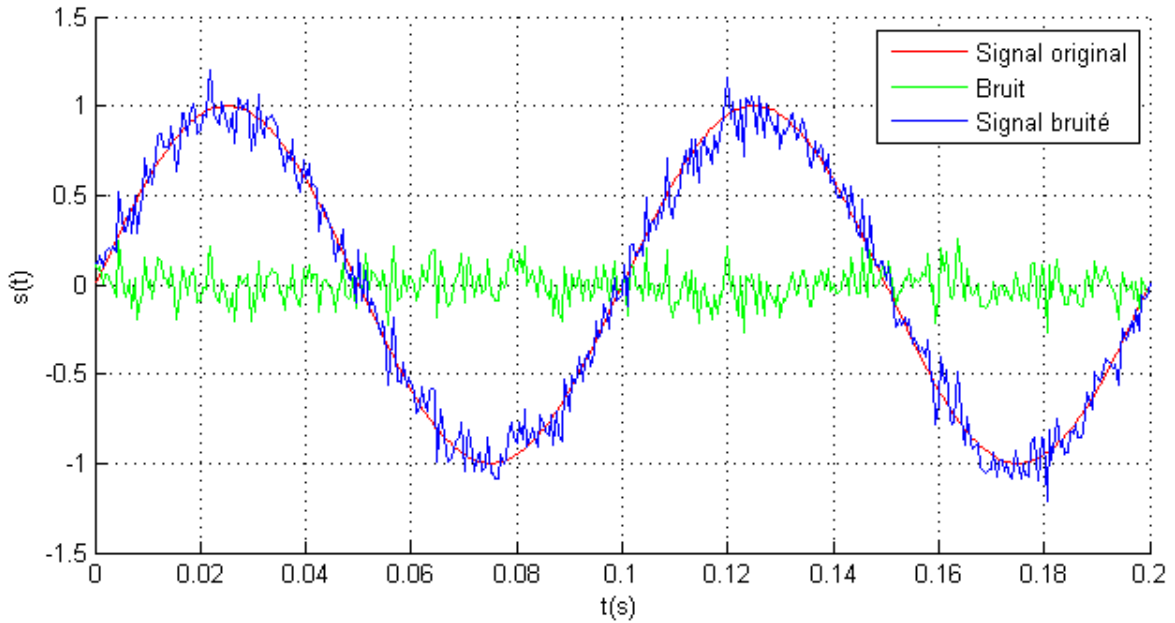
Etude de l'effet de la taille du filtre sur la qualité du signal

Pour comprendre le comportement du filtre et ses effets sur un signal. On va générer un signal sinusoïdal de fréquence F_0 , puis ajouter un bruit gaussien de la moyenne μ et de la variance σ^2 . Le signal bruité est l'addition du signal original et le bruit.

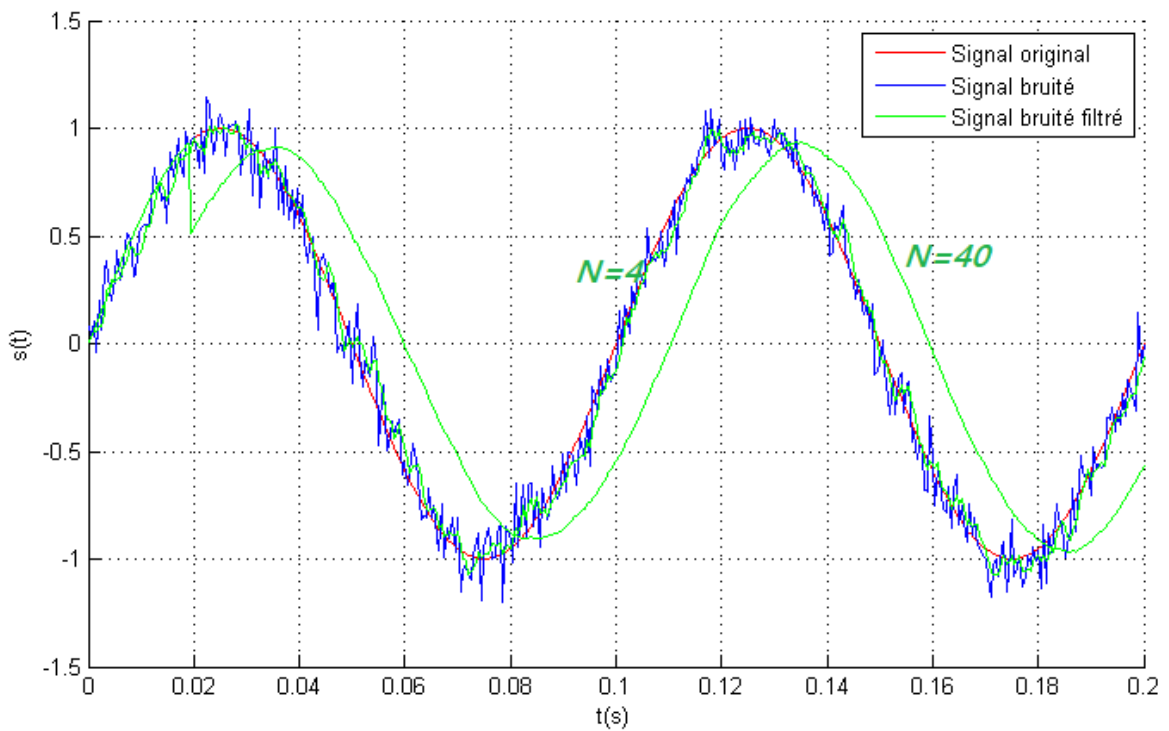
On va changer d'une façon linéaire la taille du filtre appliqué sur le signal bruité et ensuite le comparé avec le signal original.



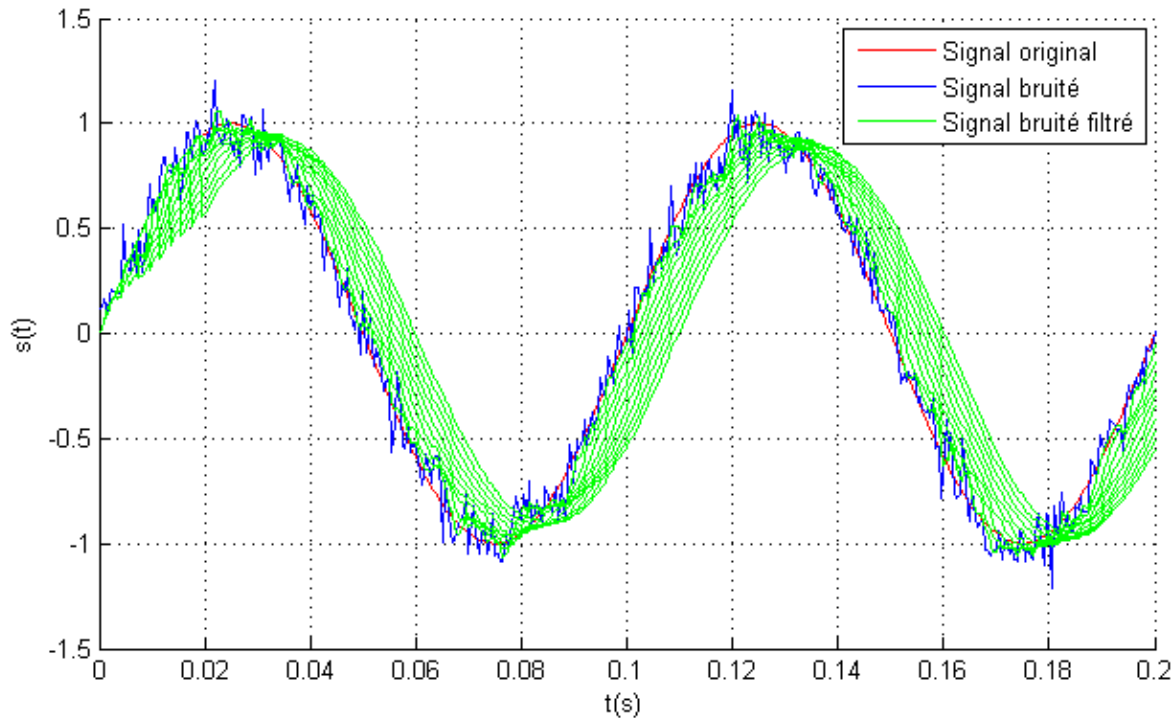
Projet électronique : Traitement du signal avec Arduino # Lissage & Seuillage d'un signal 1/3



Bruit filtré pour $N=40$ & $N=4$



Bruit filtré pour N varié de 4 à 40 avec un pas de 4



L'Erreur Quadratique Moyenne EQM ou MSE (Mean Square Error)

$$s_n^2 \equiv \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2 \text{ et } s_{n-1}^2 \equiv \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$$

L'erreur quadratique moyenne (Mean Squared Error en anglais) est l'espérance du carré de l'erreur entre la vraie valeur et sa valeur estimée.

En topométrie, on emploie plus généralement l'abrégié "EQM" pour "erreur quadratique moyenne".

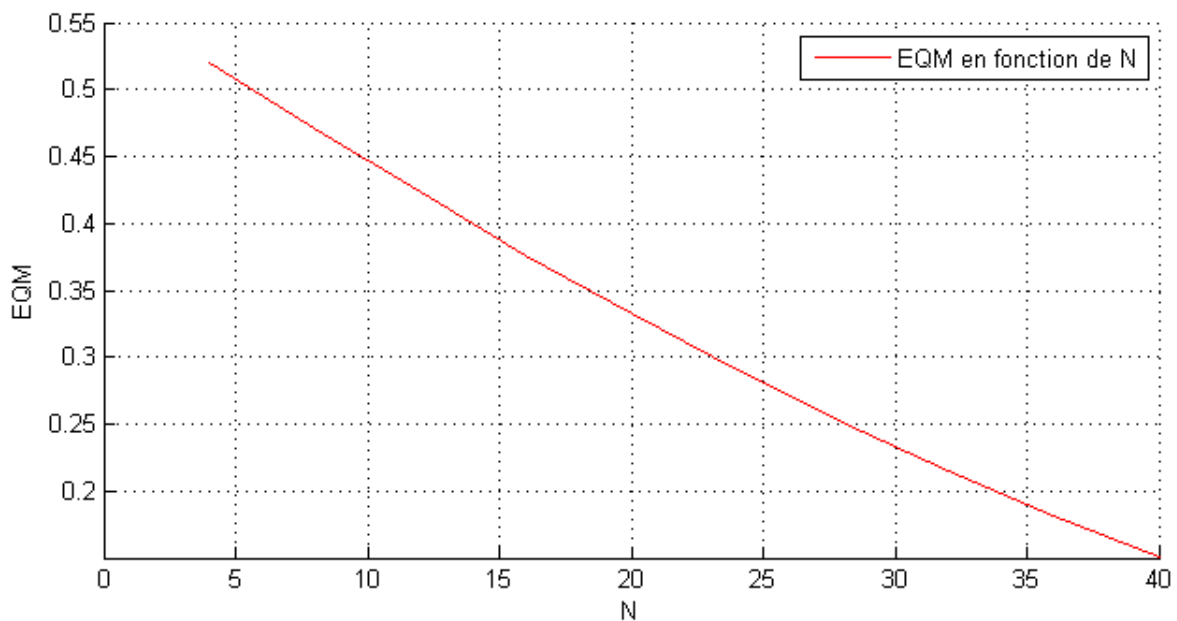
Soit un certain nombre de mesures. Chacune de ces mesures est entachée d'une erreur (rien à voir avec une faute) On appelle erreur vraie pour une mesure la différence entre la mesure et sa vraie valeur. Si cette même valeur a été mesurée plusieurs fois, on appelle erreur apparente d'une mesure isolée, la différence entre la moyenne arithmétique des mesures et la mesure isolée [Wiki].

L'EQM est une mesure statistique qui permet de mesurer la différence entre deux vecteurs.



Projet électronique : Traitement du signal avec Arduino # Lissage & Seuillage d'un signal 1/3

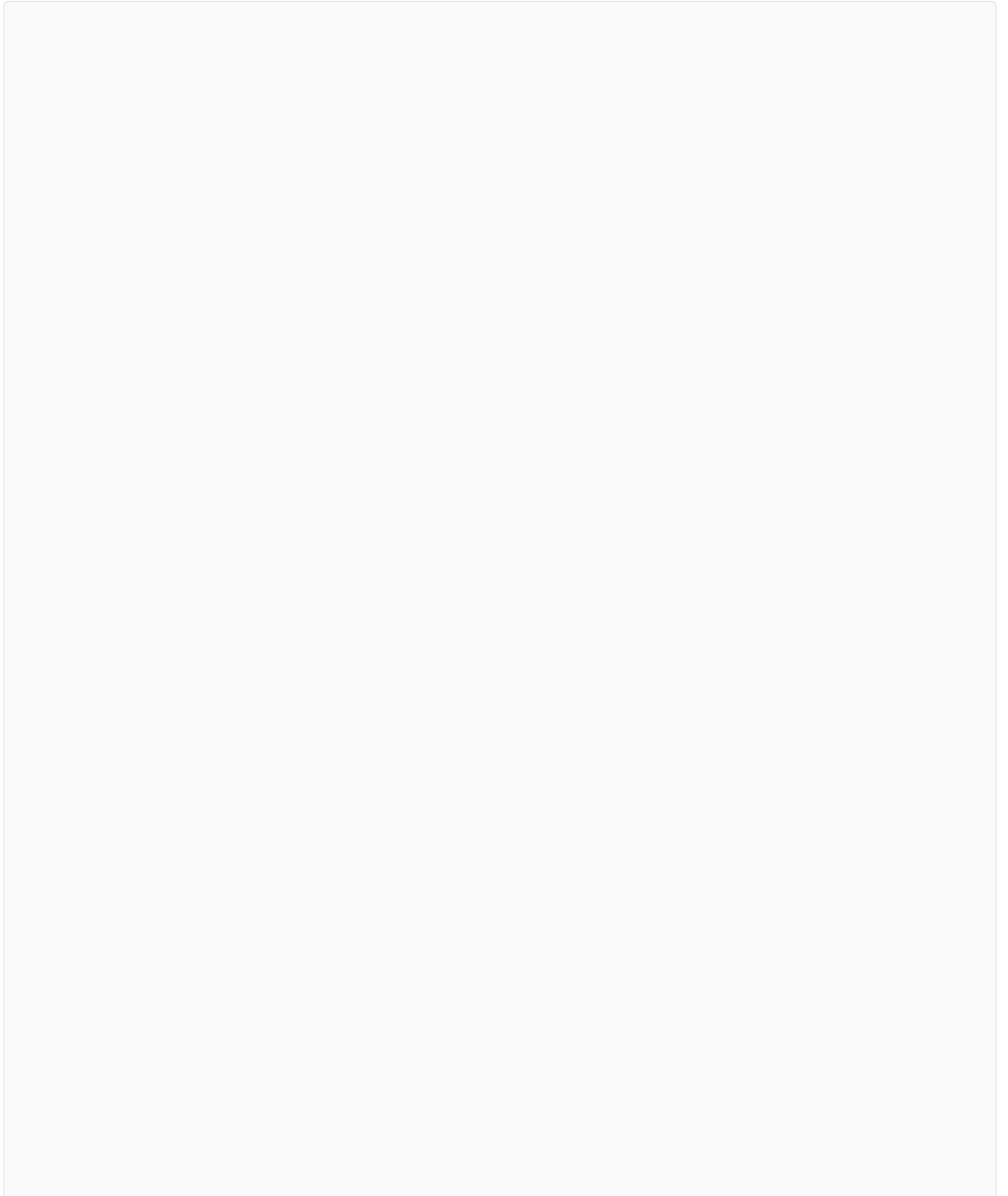
La figure ci-dessous illustre l'erreur quadratique moyenne en fonction de la taille du filtre. On constate que l'erreur décroît avec l'augmentation de la taille du filtre, donc la qualité du signal s'améliore avec l'utilisation d'un filtre de taille importante. L'inconvénient majeur du filtre, il introduit un retard de N Echantillons □



Programme Matlab



Projet électronique : Traitement du signal avec Arduino # Lissage & Seuillage d'un signal 1/3





```
% Paramètres du signal
F0=10;
T0=1/F0;
Ns=200;
Fs=F0*Ns;
Ts=1/Fs;

% Génération du signal
t= 0:Ts:2*T0;
s_t= sin(2*pi*F0*t);

% Génération du bruit
Vare = 1e-2;
Mu=0;
b_t = Mu + sqrt(Vare)*randn(1,length(s_t));

% Génération signal + bruit
s_b_t = s_t + b_t;

% Affichage
figure(1);
grid on ; hold on ;

plot(t, s_t,'r');
plot(t,b_t,'g');
plot(t,s_b_t,'b');
legend('Signal original', 'Bruit', 'Signal bruité');

xlabel('t(s)');
ylabel('s(t)');
```



```
%% Filtrage - Lissage du signal

% Paramètre du filtre moyennneur
N_filtre = 4:4:40 ;
S_filre = ones(length(N_filtre),length(s_t));
MSE=0*N_filtre;

for k=1:length(N_filtre)
    S_filre(k,:) = s_t;
    for i=N_filtre(k) : length(s_t)
        S_filre(k,i) =mean( s_b_t(i-N_filtre(k)+1:i));
    end

    % Calcul de l'erreur quadratique moyenne
    N_k= N_filtre(end);
    D = abs( s_b_t(1:end-N_k+1) - S_filre(k,N_k:end)).^2;
    MSE(k) = sum(D(:))/length(s_t);
end

% Affichage
figure(2);
grid on ; hold on ;

plot(t,s_t,'r');
plot(t,s_b_t,'b');
plot(t,S_filre,'g');
legend('Signal original','Signal bruité', 'Signal bruité filtré');

xlabel('t(s)');
ylabel('s(t)');
figure(3);
grid on ; hold on ;
```




Projet électronique : Traitement du signal avec Arduino # Lissage &
Seuillage d'un signal 1/3

```
plot(N_filtre,MSE,'r');  
legend('EQM en fonction de N');  
xlabel('N');
```



Projet électronique : Traitement du signal avec Arduino # Lissage &
Seuillage d'un signal 1/3

```
ylabel('EQM');
```



Etude et simulation numérique du filtre Médian avec Matlab

Le filtre médian est un filtre numérique simple, souvent utilisé pour la réduction de bruit. La réduction de bruit est une étape de post-traitement classique visant à améliorer les résultats de traitements futurs du signal. La technique de filtre médian est largement utilisée en traitement d'images numériques et traitement des signaux en général car il permet sous certaines conditions de réduire le bruit dans le signal.

Le principe du filtre est très simple ! Il consiste à trier un morceau de N échantillons du signal (par ordre croissant ou décroissant), puis sélectionner la valeur médiane du tableau, c.à.d. la valeur de l'indice $N/2$! Alors la valeur médiane est égale à $\text{TabSort}[N/2]$.

1- Le signal Original

12	11	10	45	21	36	40	24	20	14	5	64	12	45
0	1	2	3	4	5	6	7	8	9	10	11	12	13

2- Morceau du signal sur N=7 - Sig

12	11	10	45	21	36	40
----	----	----	----	----	----	----

3- Tri du signal par ordre croissant SigSort

10	11	12	21	36	40	45
----	----	----	----	----	----	----

4- La valeur médiane =21 = SigSort $[(N-1)/2]$

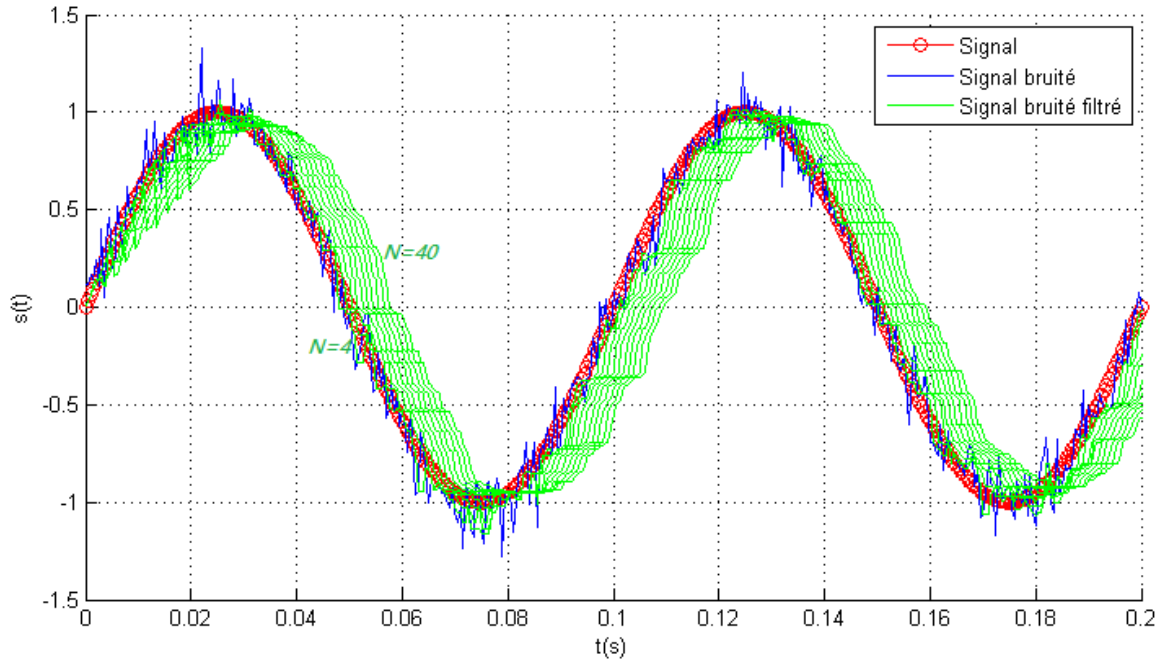
10	11	12	21	36	40	45
----	----	----	----	----	----	----

Résultats des simulations avec Matlab

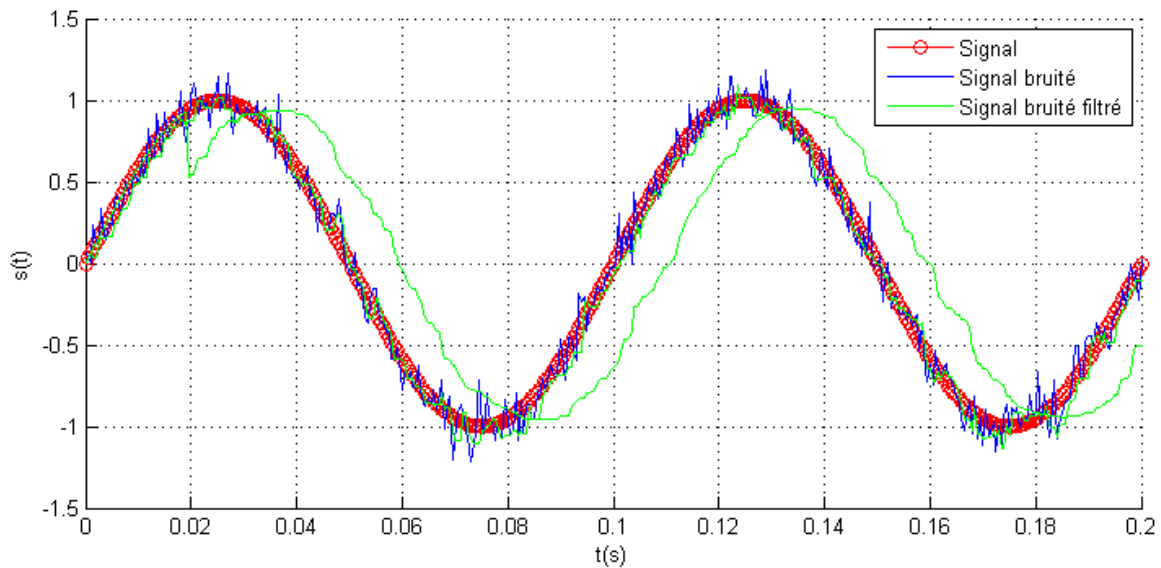
Bruit filtré pour N varié de 4 à 40 avec un pas de 4



Projet électronique : Traitement du signal avec Arduino # Lissage & Seuillage d'un signal 1/3



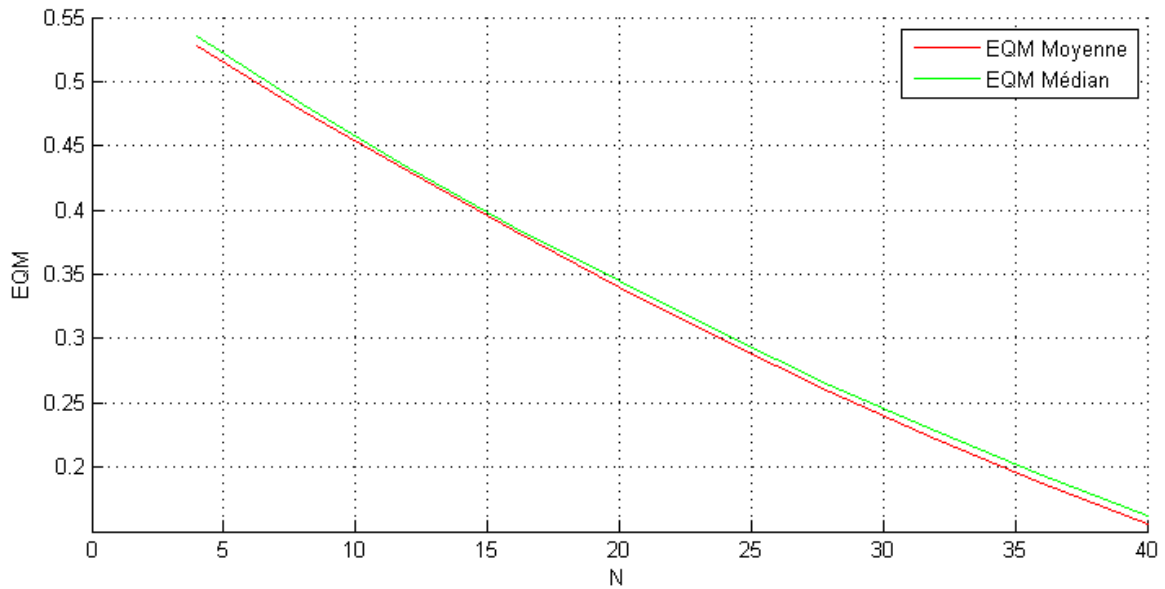
Bruit filtré pour $N=40$ & $N=4$



On constate que le filtre médian permet d'obtenir des bons résultats. On voit clairement que la qualité du signal filtré est dégradée par rapport au filtre moyenneur. La figure ci-dessous illustre la comparaison entre les EQM des deux filtres.



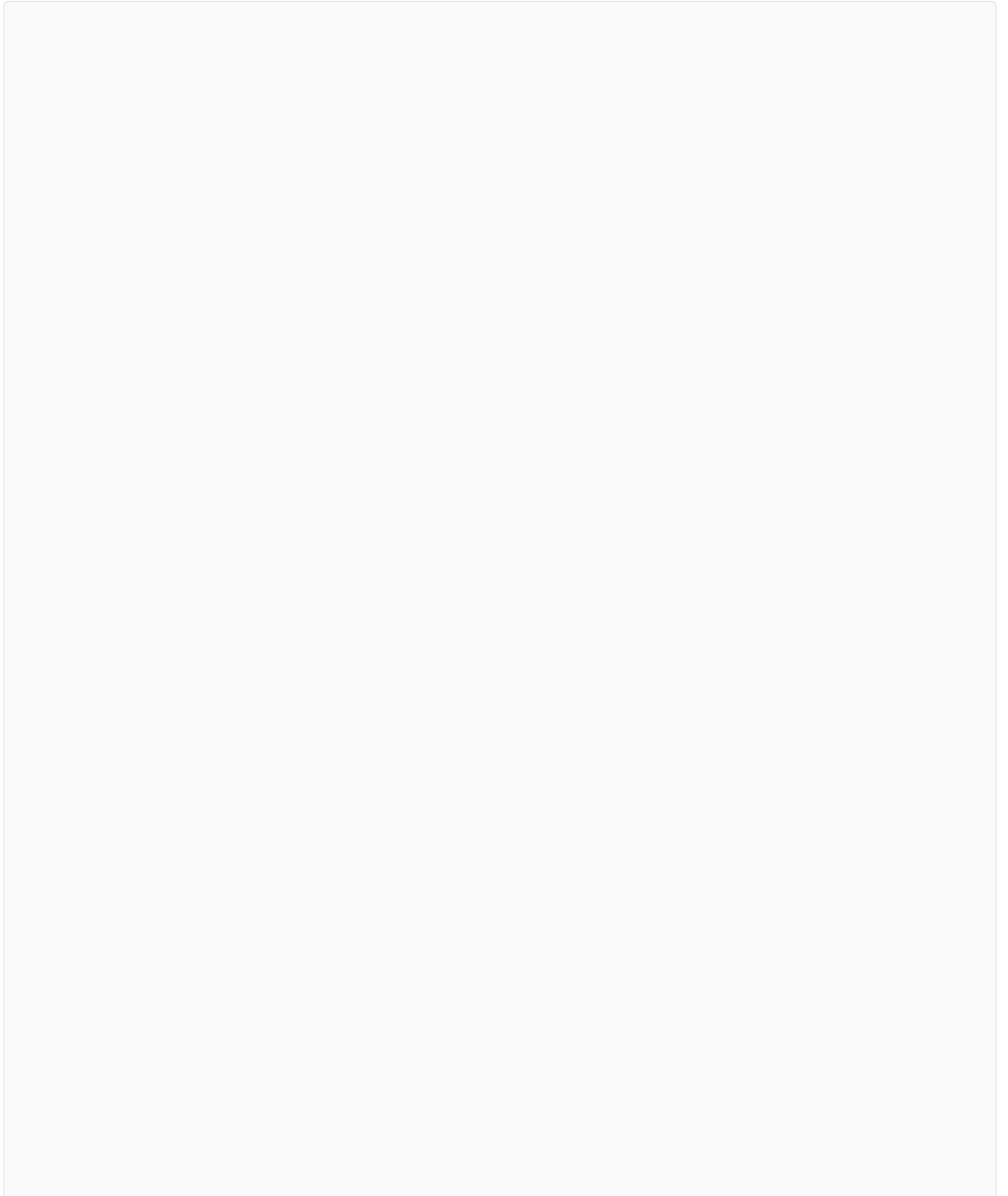
Projet électronique : Traitement du signal avec Arduino # Lissage & Seuillage d'un signal 1/3



Programme Matlab



Projet électronique : Traitement du signal avec Arduino # Lissage & Seuillage d'un signal 1/3





```
% Paramètres du signal
F0=10;
T0=1/F0;
Ns=200;
Fs=F0*Ns;
Ts=1/Fs;

% Génération du signal
t= 0:Ts:2*T0;
s_t= sin(2*pi*F0*t);

% Génération du bruit
Vare = 1e-2;
Mu=0;
b_t = Mu + sqrt(Vare)*randn(1,length(s_t));

% Génération signal + bruit
s_b_t = s_t + b_t;

% Paramètres du filtre Médian
N_filtre = 4:4:40 ;
S_filre = ones(length(N_filtre),length(s_t));
MSE2=0*N_filtre;

for k=1:length(N_filtre)
    S_filre(k,:) = s_t;
    for i=N_filtre(k) : length(s_t)
        MedTab= sort(s_b_t(i-N_filtre(k)+1:i));
        S_filre(k,i) =MedTab(round(N_filtre(k)/2.0));
    end
end
```



```
% Calcul de l'erreur quadratique moyenne
N_k= N_filtre(end);
D = abs( s_b_t(1:end-N_k+1) - S_filre(k,N_k:end)).^2;
MSE2(k) = sum(D(:))/length(s_t);

end

% Affichage
figure(1);
grid on ; hold on ;

plot(t,s_t,'r-o');
plot(t,s_b_t,'b');
plot(t,S_filre,'g');
legend('Signal','Signal bruité', 'Signal bruité filtré');

xlabel('t(s)');
ylabel('s(t)');
figure(2);
grid on ; hold on ;
plot(N_filtre,MSE2,'g');
legend('EQM en fonction de N');
xlabel('N');
```



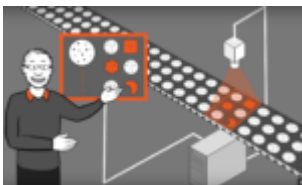

Projet électronique : Traitement du signal avec Arduino # Lissage &
Seuillage d'un signal 1/3

```
ylabel('EQM');
```



Un petit commentaire de vous, un Grand encouragement pour nous ☐

Dernières réalisations



C'e

st quoi le traitement
d'image | What Is Image
Processing ?



Intr

oduction au filtrage
numérique



Ard

uino #40: le signal et le
bruit - le sinus cardinal



Proj

et commander une carte



Projet électronique : Traitement du signal avec Arduino # Lissage & Seuillage d'un signal 1/3

Arduino avec
Smartphone (sans
Modem)



Proj

et électronique :
Oscilloscope numérique
à base du
microcontrôleur
PIC18F4680 à liaison
série RS232



Ard

uino #4: la boucle for



C'e

st quoi un Hacheur #1 ?



C'e

st quoi un transistor ? |

How transistors work ?



Ard

uino #17: les 4 qualités
d'une variable



Articles

- [Arduino #34: PWM Comment mesurer la valeur moyenne et la valeur efficace d'un signal PWM](#)
- [Projet ordonnancement des tâches avec Arduino](#)
- [Arduino #33: Comment mesurer le temps d'exécution - deux techniques](#)
- [Projet commander une carte Arduino avec Smartphone \(sans Modem\)](#)
- [Projet lampe tactile avec Arduino](#)
- [Arduino #18: les mémoires](#)
- [Arduino #4: la boucle for](#)