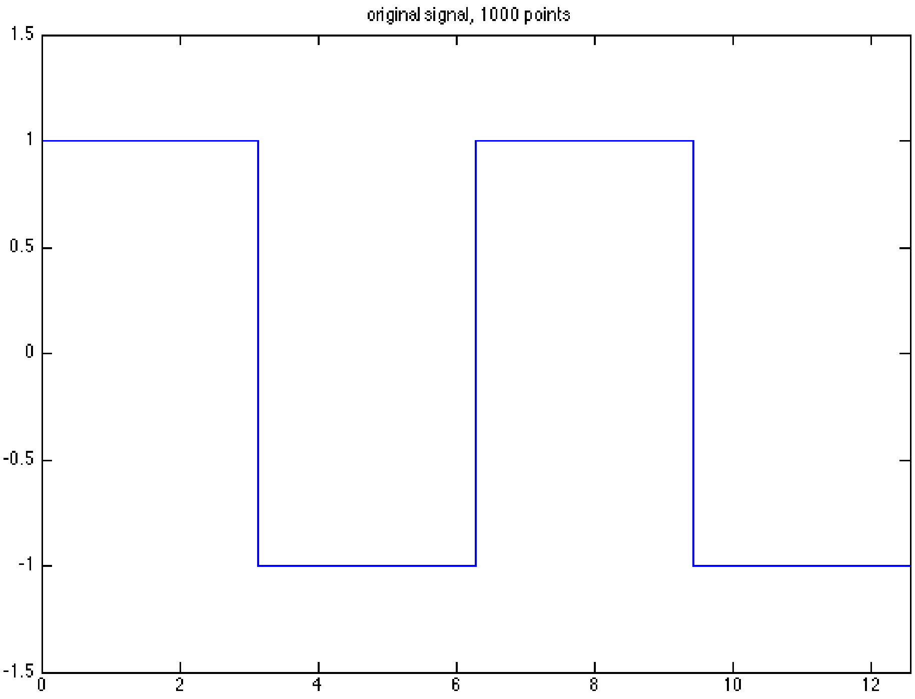
Table of Contents

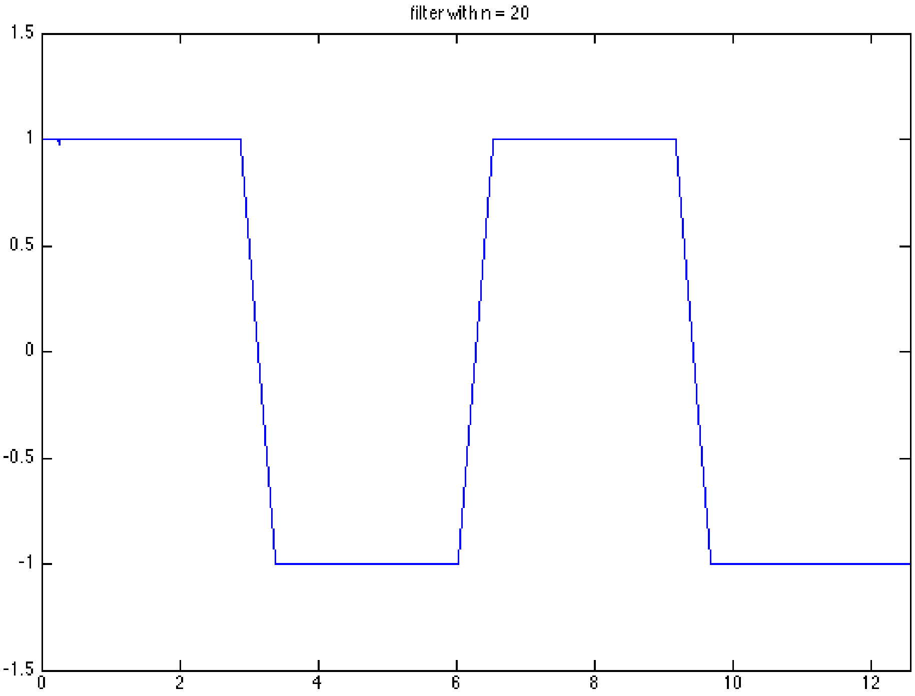
```
% digital filtering (smoothing) of signals

% This can be a deep topic but here we will look at a simple case.
% We replace each signal value by the
% mean of it and its n preceding and n following neigboring
% values, where n is an integer.
% The larger the value of n, the more filtering or smoothing.
% This filter is often used to reduce random noise on a signal.
% If n is too large, the "shape" of the underlying signal
% can be affected.
% In NANO 15, HW 7 in 2012, we used this filter on 2D arrays of
% color values to "blur" a region of a JPG image, that is,
% to reduce the contrast or difference between neighboring pixels.
% Below, we illustrate using this filter on 1D arrays of values.
% Also see
% http://en.wikipedia.org/wiki/Digital_filter

clear all
```

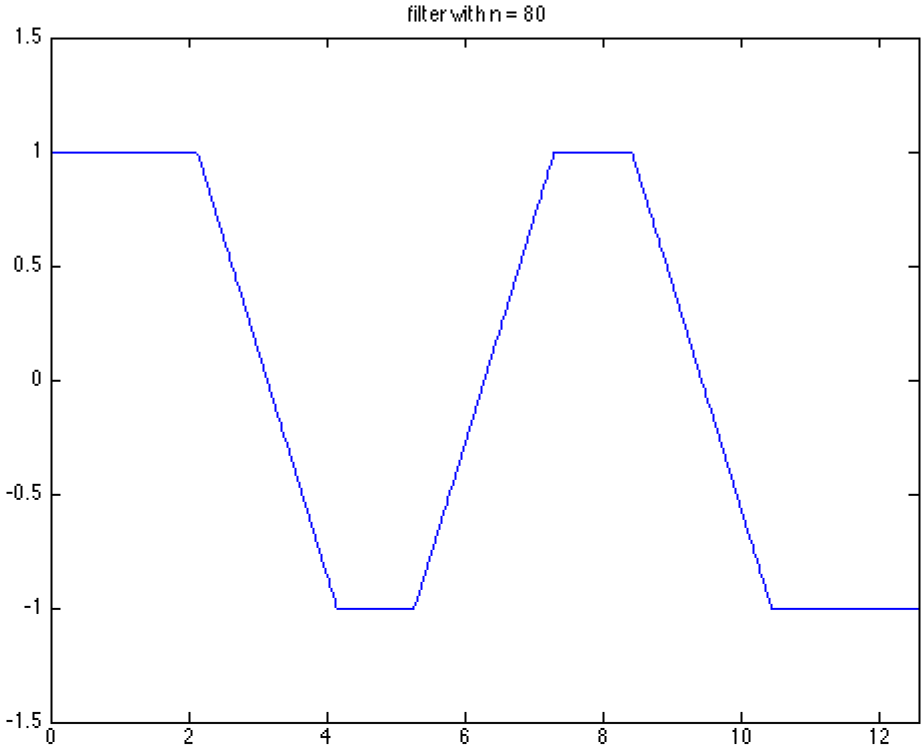
filter square wave

```
np = 1000;
x = linspace(0,4*pi,np);
y = sin(x);
y = sign(y); % convert sine wave to square wave
figure(1), plot(x,y), axis([0 4*pi -1.5 1.5])
tt = sprintf('original signal, %i points',np);
title(tt)
% now filter
n = 20; % number before and after a point to use in average
ys = y;
for i = 1+n:length(x)-n
    ys(i) = mean(y(1,i-n:i+n));
end
figure(2), plot(x,ys), axis([0 4*pi -1.5 1.5])
tt = sprintf('filter with n = %i',n);
title(tt)
% now filter with larger n
n = 80;
ys = y;
for i = 1+n:length(x)-n
    ys(i) = mean(y(1,i-n:i+n));
end
```

```
figure(3), plot(x,ys), axis([0 4*pi -1.5 1.5])
tt = sprintf('filter with n = %i',n);
title(tt)
```

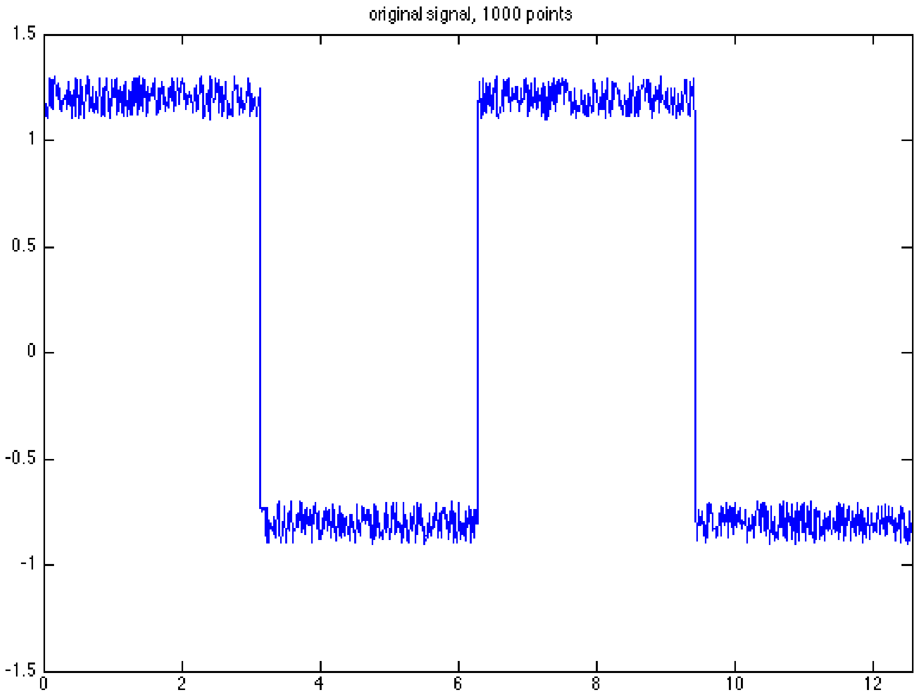




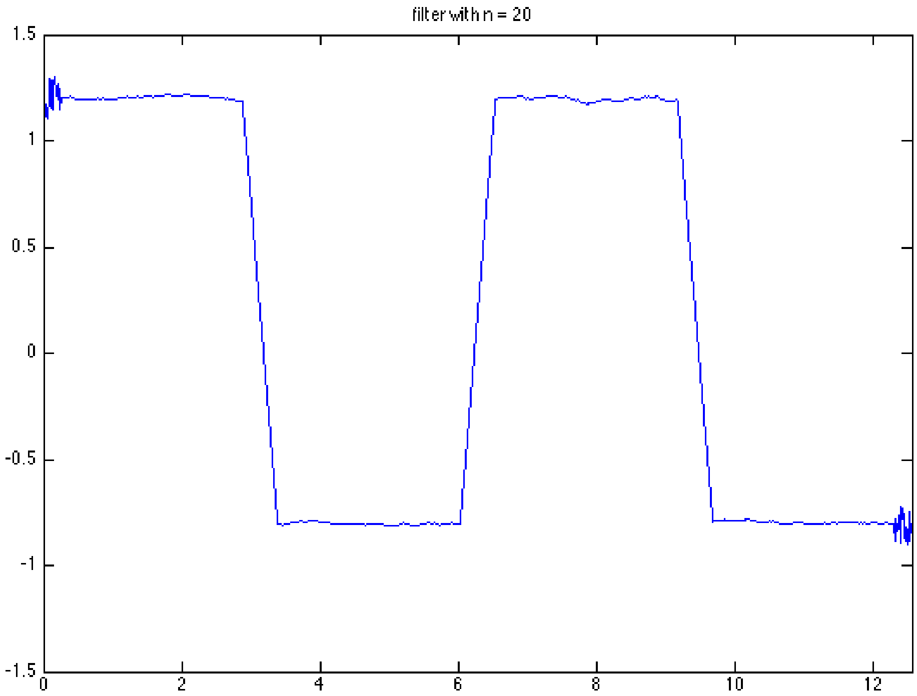


filter square wave with noise

```
sd = 0.2;
y = y + sd*(rand(1,length(x)) + 0.5); % add random noise
figure(1), plot(x,y), axis([0 4*pi -1.5 1.5])
tt = sprintf('original signal, %i points',np);
title(tt)
% now filter
n = 20; % number before and after a point to use in average
ys = y;
for i = 1+n:length(x)-n
    ys(i) = mean(y(1,i-n:i+n));
end
figure(2), plot(x,ys), axis([0 4*pi -1.5 1.5])
tt = sprintf('filter with n = %i',n);
title(tt)
```



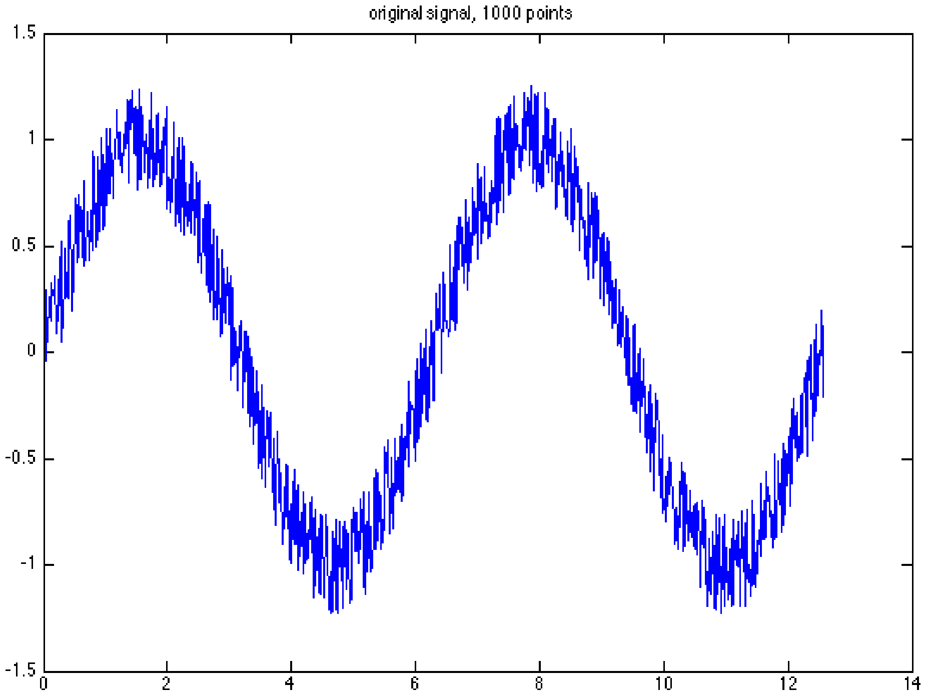
original signal, 1000 points

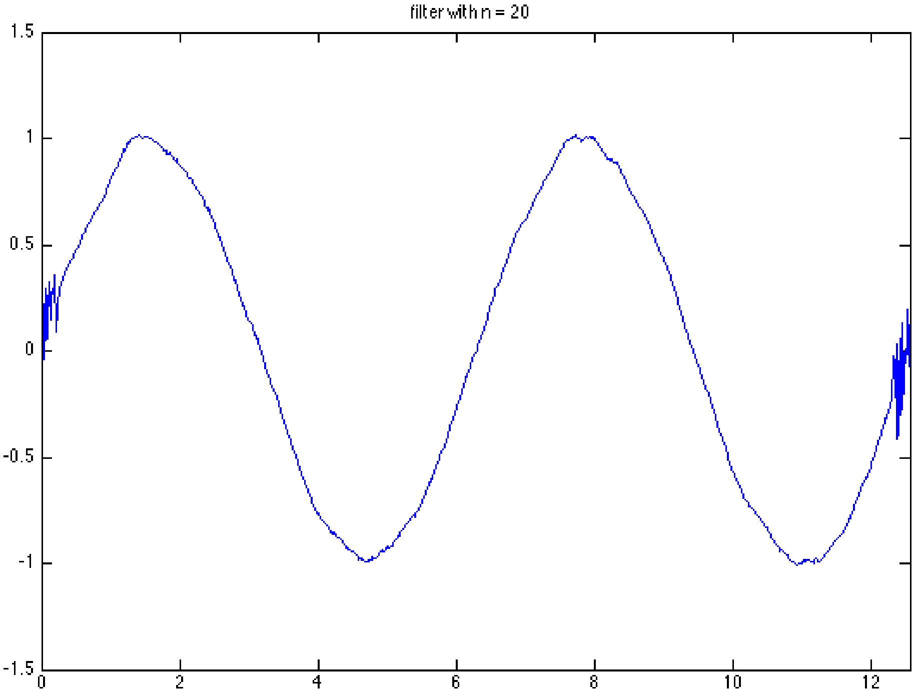


filter with n = 20

filter sine wave with noise

```
y = sin(x); % go back to sine wave
sd = 0.5;
y = y + sd*(rand(1,length(x)) - 0.5); % add random noise
figure(1), plot(x,y)
tt = sprintf('original signal, %i points',np);
title(tt)
% now filter
n = 20; % number before and after a point to use in average
ys = y;
for i = 1+n:length(x)-n
    ys(i) = mean(y(1,i-n:i+n));
end
figure(2), plot(x,ys), axis([0 4*pi -1.5 1.5])
tt = sprintf('filter with n = %i',n);
title(tt)
```





Published with MATLAB® 8.0