https://www.halvorsen.blog

Low-pass Filter

Hans-Petter Halvorsen

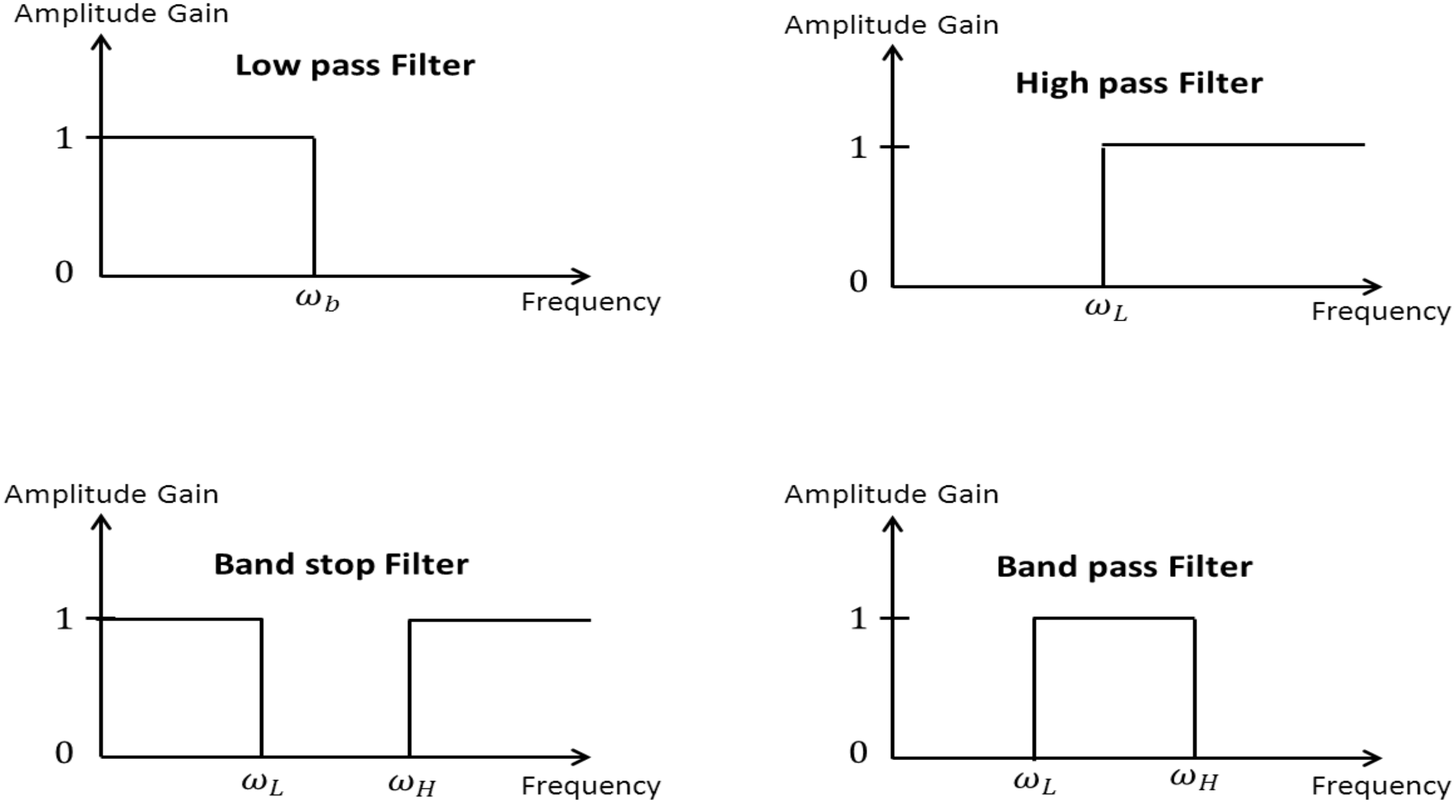
Contents

- Introduction to Filters
  - Overview of different Filters
  - What is a Low-pass Filter?
  - Why do we need a Lowpass Filter?
- Using a built-in Lowpass Filter in LabVIEW
- Create your own Lowpass Filter from scratch

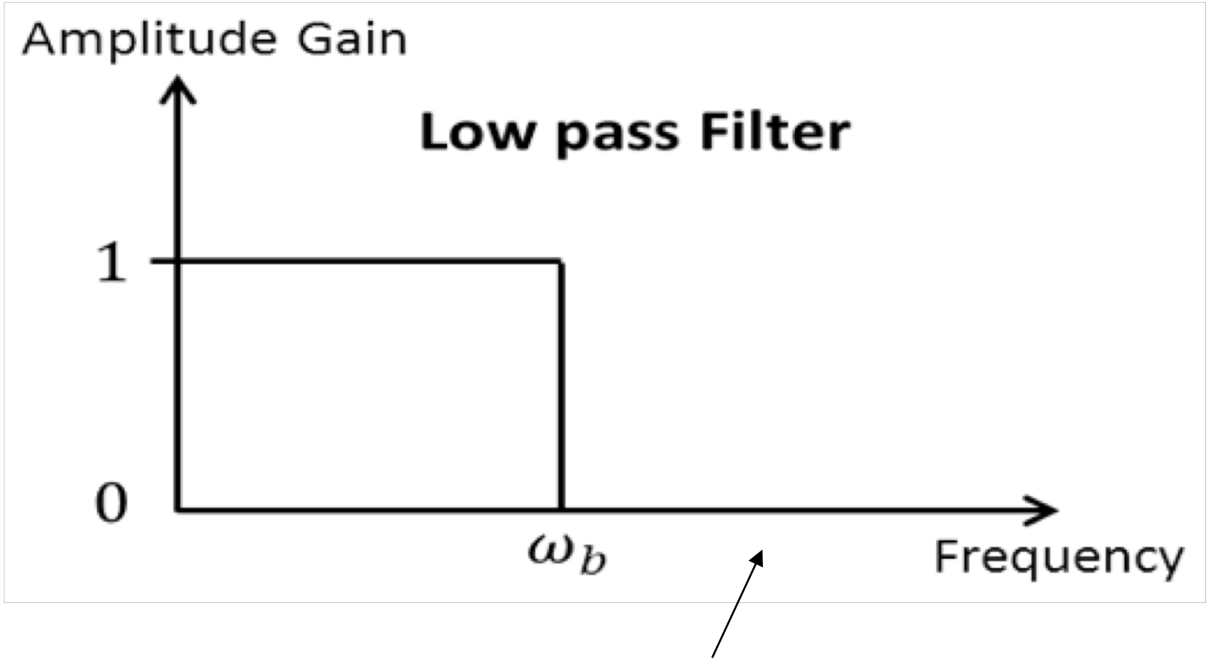
Filters

- A Filters are typically used in frequency response analysis
- A filter is used to remove given frequencies or an interval of frequencies from a signal.
- Such an application would typically be to remove noise from a signal.
- The most common is the low pass filter.
- We have 4 types of filter:
  - Low-pass Filter
  - High-pass Filter
  - Band-pass Filter
  - Band-stop Filter

Filters

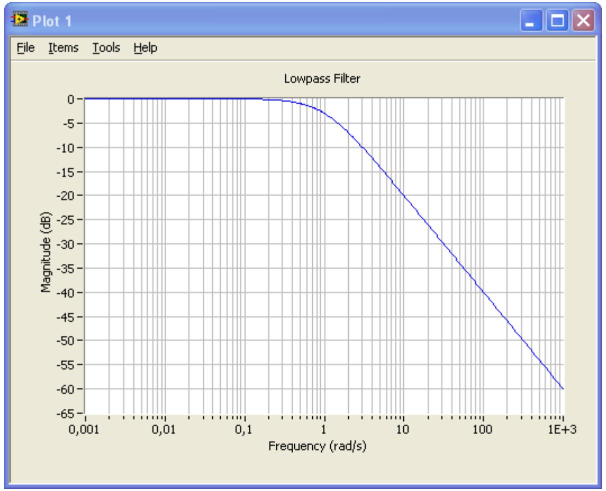

Amplitude Gain
Low pass Filter
1
0
ω_b
Frequency
Amplitude Gain
High pass Filter
1
0
ω_L
Frequency
Amplitude Gain
Band stop Filter
1
0
ω_L
ω_H
Frequency
Amplitude Gain
Band pass Filter
1
0
ω_L
ω_H
Frequency

Low-pass Filter



High frequencies (above ω_b) are removed (or attenuated)

$$H(s) = \frac{1}{Ts+1} = \frac{1}{\frac{1}{\omega_b}s+1}$$



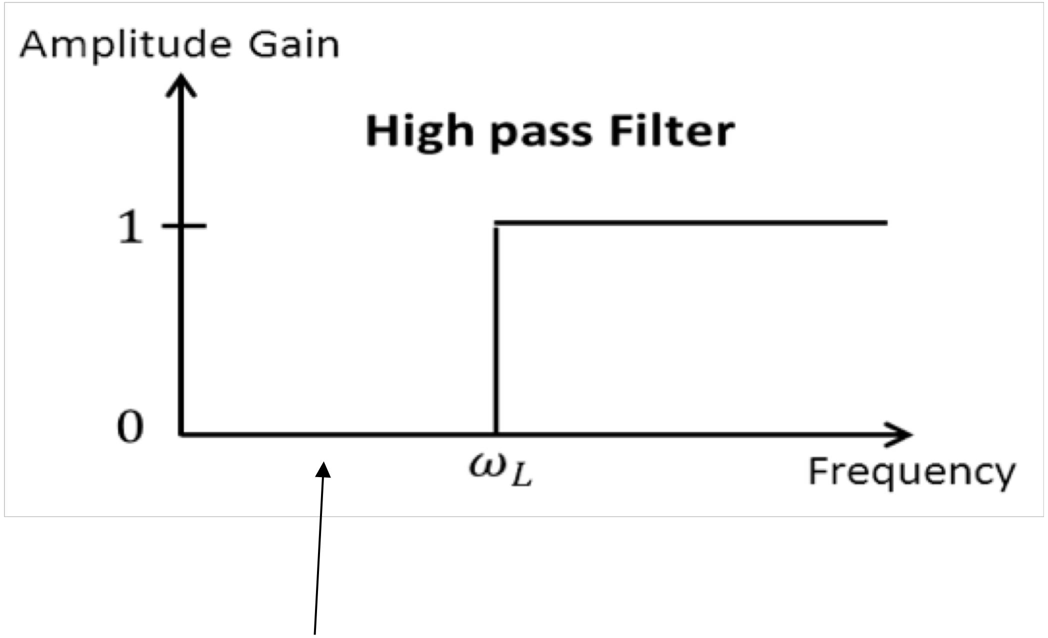
Low-pass Filter in LabVIEW

Low-pass Filter

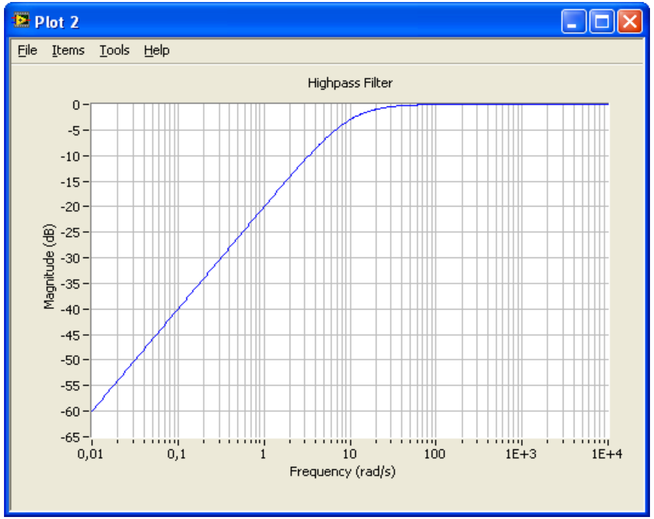
- In Measurement systems and Control Systems we typically need to deal with noise
- Noise is something we typically don’t want
- Noise is high-frequency signals
- Low-pass Filters are used to remove noise from the measured signals

High-pass Filter

$$H(s) = \frac{Ts}{Ts + 1} = \frac{\frac{1}{\omega_L}s}{\frac{1}{\omega_L}s + 1}$$

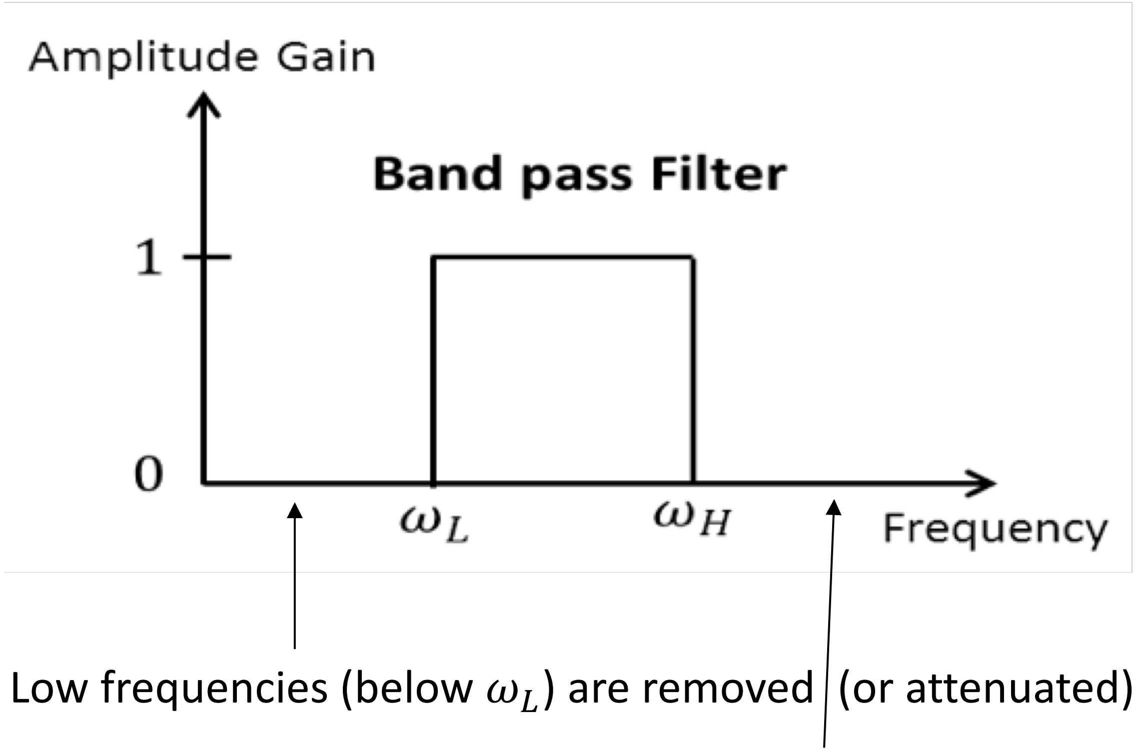


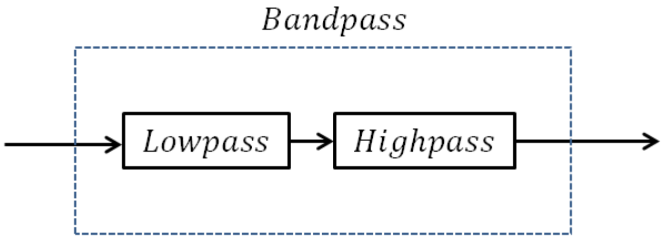
Low frequencies (below ω_b) are removed (or attenuated)

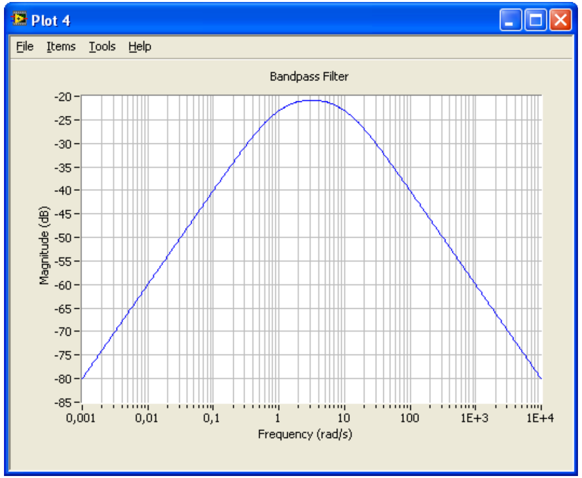


High-pass Filter in LabVIEW

Band-pass Filter





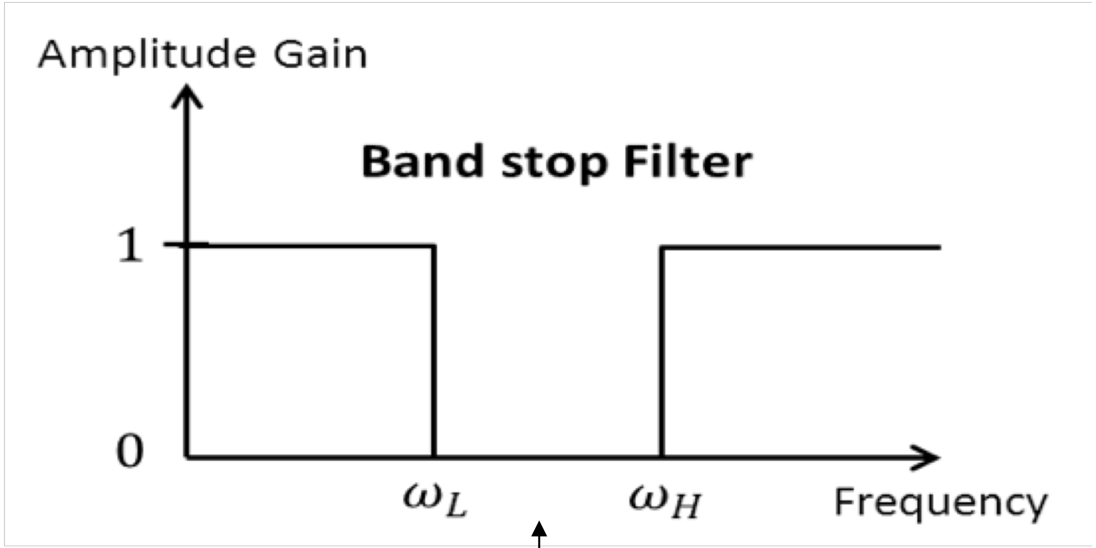


Low frequencies (below ω_L) are removed (or attenuated)

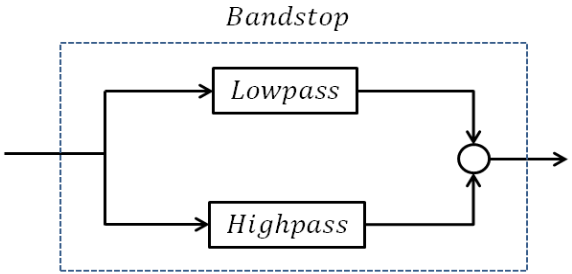
and High frequencies (above ω_H) are removed (or attenuated)

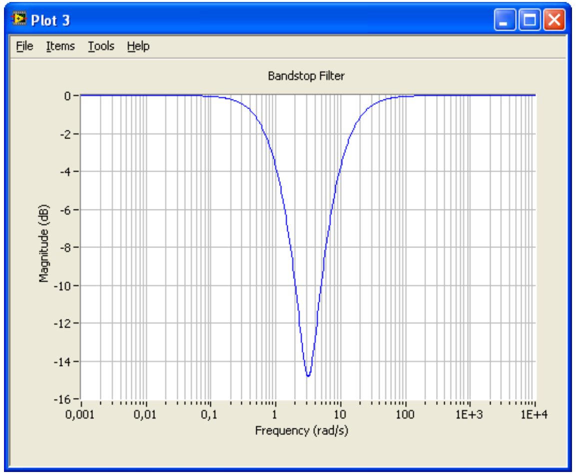
Band-pass Filter in LabVIEW

Band-stop Filter



Frequencies between ω_L and ω_H are removed





Band-stop Filter in LabVIEW

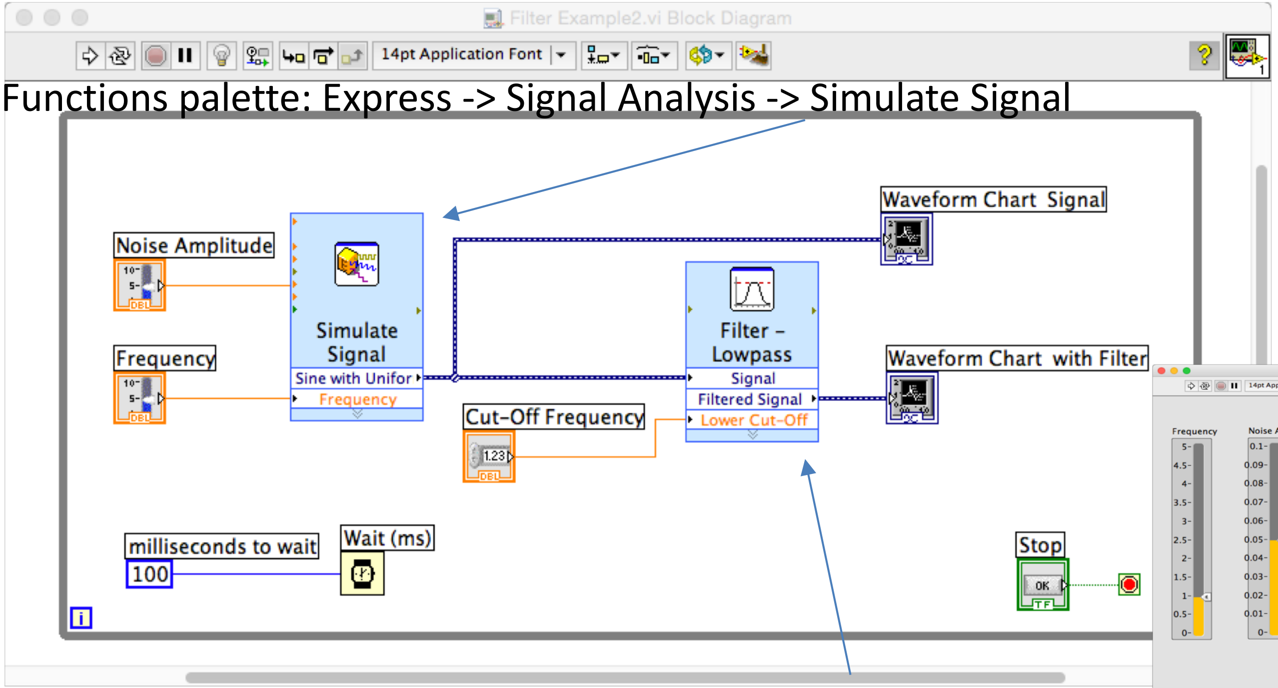
https://www.halvorsen.blog

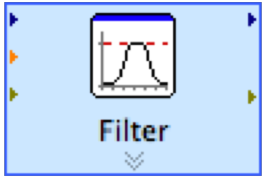
Using a built-in Low-pass Filter in LabVIEW

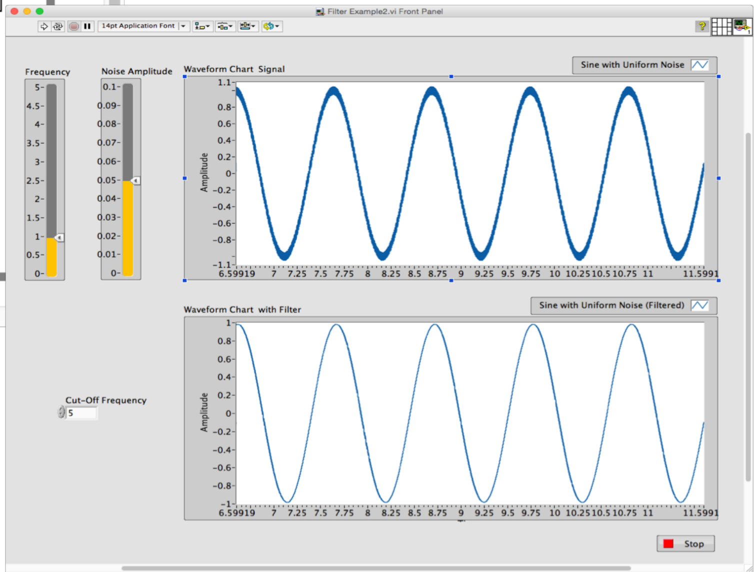
Hans-Petter Halvorsen

Using a Low-pass Filter to reduce Noise

Functions palette: Express -> Signal Analysis -> Simulate Signal



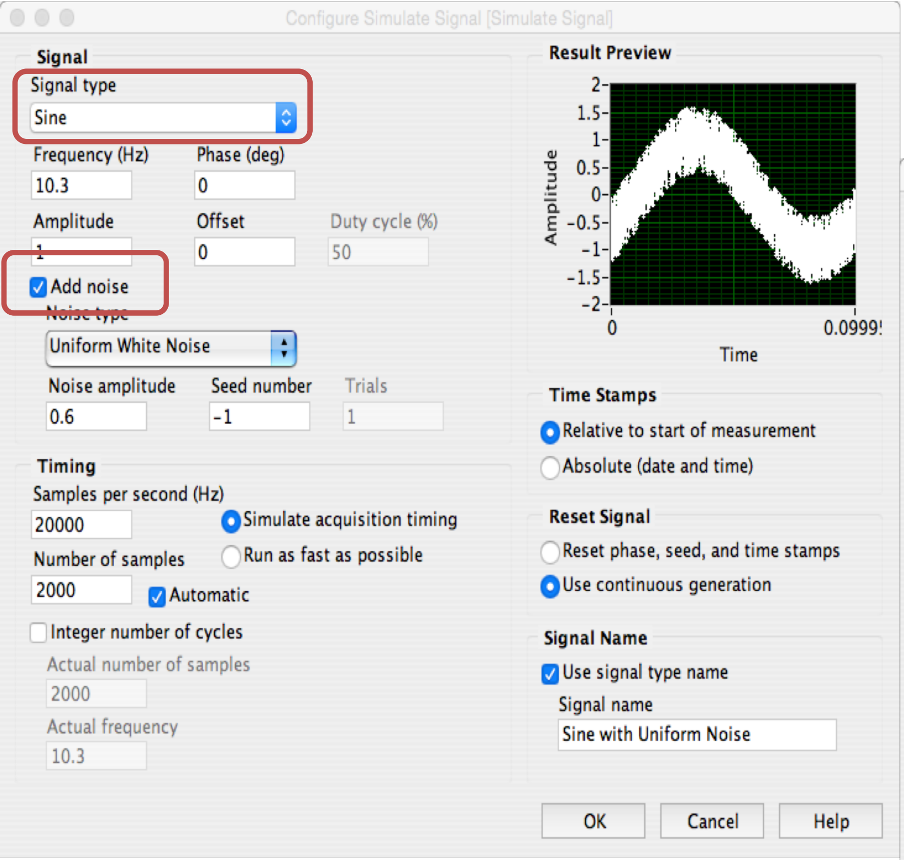


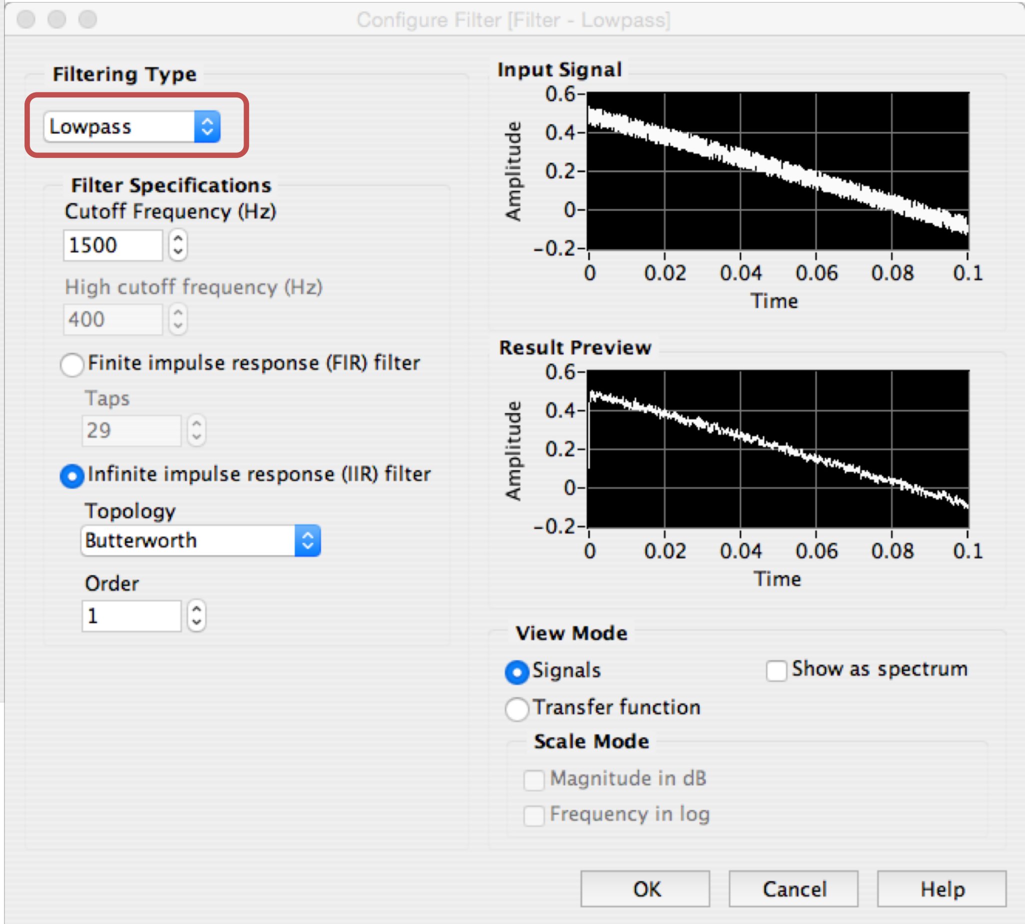


Functions palette: Express -> Signal Analysis -> Filter

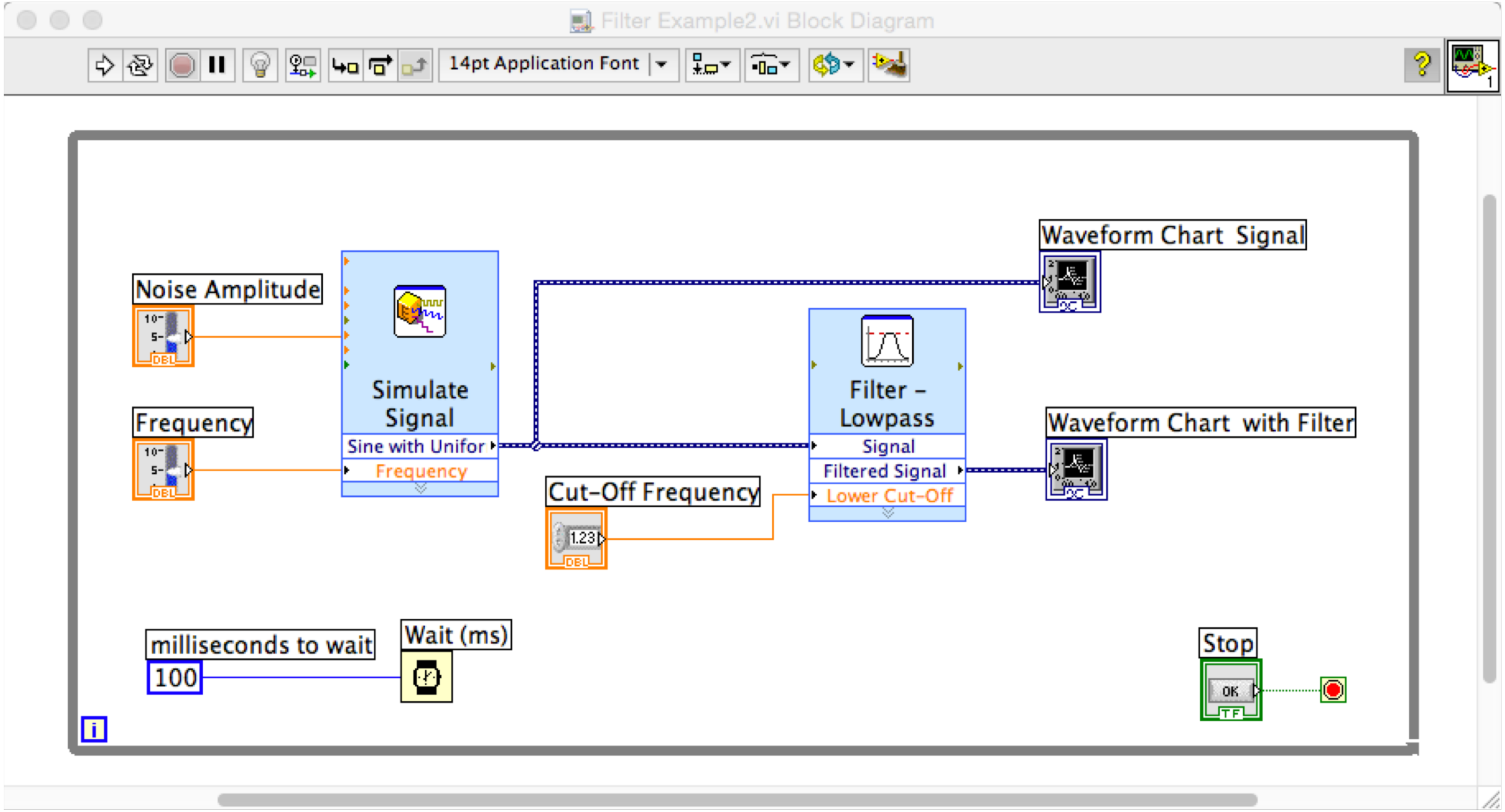
Here we use one of the built-in (Low-pass) Filters

Properties

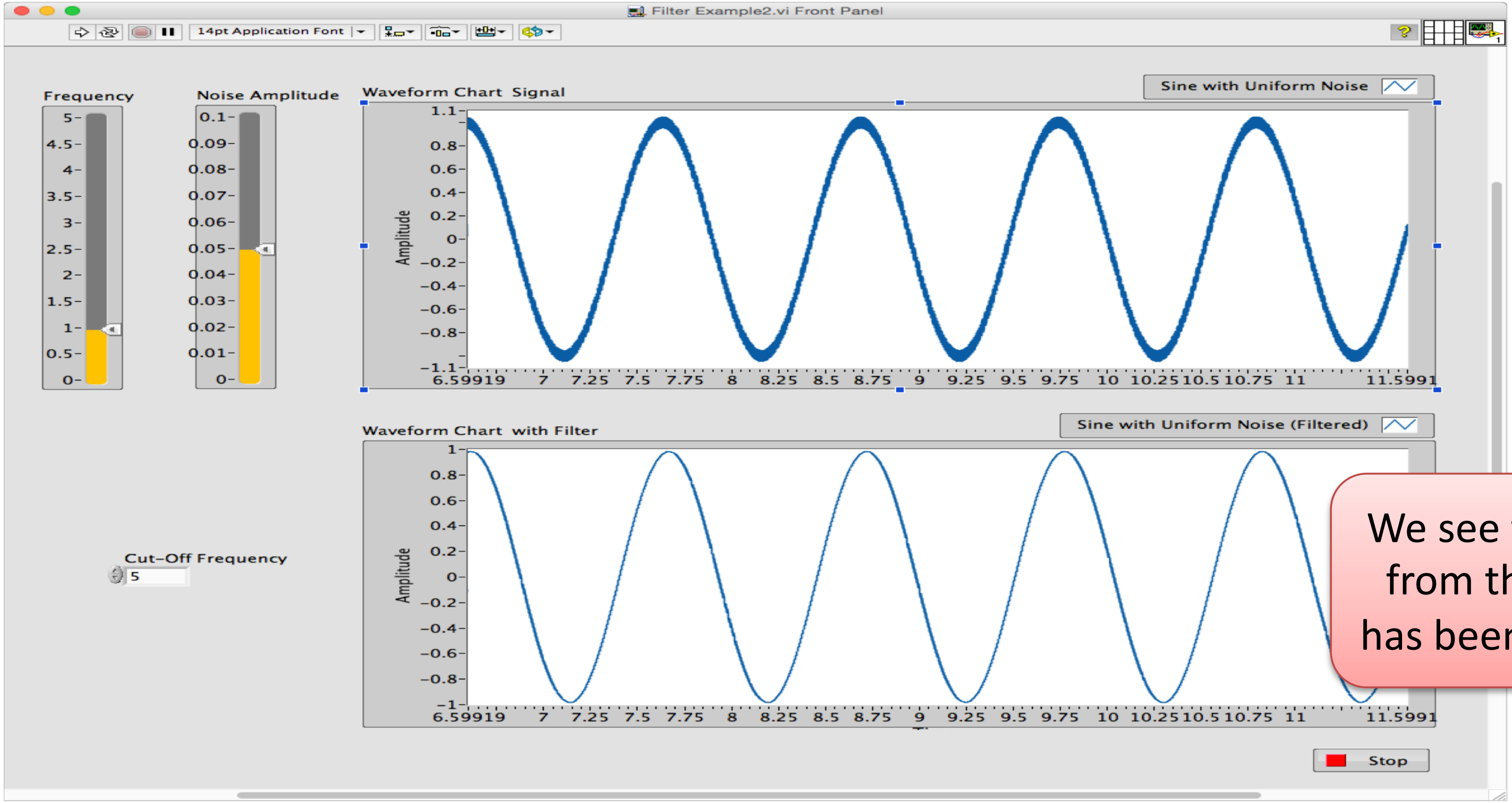

Configure Simulate Signal [Simulate Signal]
Signal
Signal type
Sine
Frequency (Hz)
10.3
Phase (deg)
0
Amplitude
1
Offset
0
Duty cycle (%)
50
Add noise
Noise type
Uniform White Noise
Noise amplitude
0.6
Seed number
-1
Trials
1
Timing
Samples per second (Hz)
20000
Simulate acquisition timing
Run as fast as possible
Number of samples
2000
Automatic
Integer number of cycles
Actual number of samples
2000
Actual frequency
10.3
Result Preview
Amplitude
Time
Time Stamps
Relative to start of measurement
Absolute (date and time)
Reset Signal
Reset phase, seed, and time stamps
Use continuous generation
Signal Name
Use signal type name
Signal name
Sine with Uniform Noise
OK
Cancel
Help


Configure Filter [Filter - Lowpass]
Filtering Type
Lowpass
Filter Specifications
Cutoff Frequency (Hz)
1500
High cutoff frequency (Hz)
400
Finite impulse response (FIR) filter
Taps
29
Infinite impulse response (IIR) filter
Topology
Butterworth
Order
1
Input Signal
Amplitude
Time
Result Preview
Amplitude
Time
View Mode
Signals
Show as spectrum
Transfer function
Scale Mode
Magnitude in dB
Frequency in log
OK
Cancel
Help

Using a Low-pass Filter to reduce Noise



Using a Low-pass Filter to reduce Noise



We see the noise from the signal has been reduced

https://www.halvorsen.blog

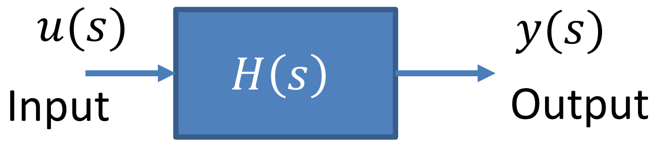
Create your own Low-pass Filter from scratch

Hans-Petter Halvorsen

Low-pass Filter

A Low-pass Filter has the following Transfer Function:

$$H(s) = \frac{y(s)}{u(s)} = \frac{1}{T_f s + 1}$$



In LabVIEW we can implement a Low-pass Filter in many ways.
If we want to implement the Low-pass Filter in a text-based programming or using e.g., the Formula Node in LabVIEW we typically need to find a discrete version of the filter.

Low-pass Filter

A Low-pass Filter has the following Transfer Function

$$H(s) = \frac{y(s)}{u(s)} = \frac{1}{T_f s + 1}$$

We can find the Differential Equation for this filter using Inverse Laplace

We get:

$$y(s)\left[T_f s + 1\right] = u(s)$$

$$T_f y(s)s + y(s) = u(s)$$

Finally we get the following differential equation:

$$T_f \dot{y} + y = u$$

We apply Euler on the Differential Equation in order to find the Discrete Differential equation

Discretization of Low-pass Filter

We have the following differential equation:

$$T_f \dot{y} + y = u$$

We use Euler Backward method: $\dot{x} \approx \frac{x(k)-x(k-1)}{T_s}$

Then we get:

$$T_f \, \frac{y(k) - y(k-1)}{T_s} + y(k) = u(k)$$

This gives: $y(k) = \frac{T_f}{T_f+T_s} y(k-1) + \frac{T_s}{T_f+T_s} u(k)$

We define:

$$\frac{T_s}{T_f + T_s} \equiv a$$

This finally gives:

$$y(k) = (1-a)y(k-1) + au(k)$$

This equation can easily be implemented in LabVIEW or another programming language

Discrete Low-pass Filter Example

Lowpass Filter Transfer function:

$$H(s) = \frac{y(s)}{u(s)} = \frac{1}{T_f s + 1}$$

Inverse Laplace the differential Equation:

$$T_f \dot{y} + y = u$$

We use the Euler Backward method:

$$\dot{x} = \frac{x_k - x_{k-1}}{T_s}$$

This gives:

$$T_f \frac{y_k - y_{k-1}}{T_s} + y_k = u_k$$

$$y_k = \frac{T_f}{T_f + T_s} y_{k-1} + \frac{T_s}{T_f + T_s} u_k$$

We define:

$$\frac{T_s}{T_f + T_s} \equiv a$$

This gives:

$$y_k = (1 - a) y_{k-1} + a u_k$$

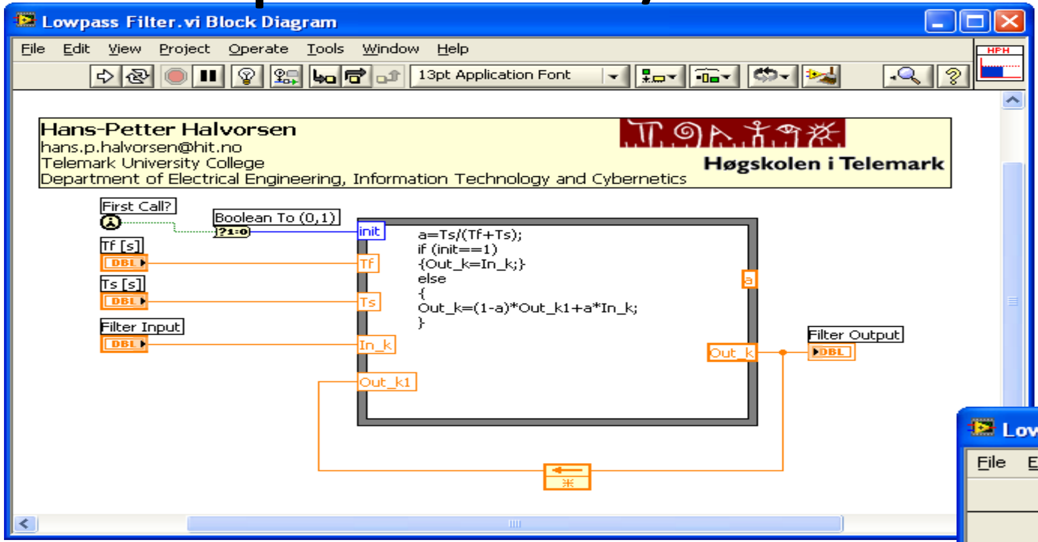
Filter output (y_k)

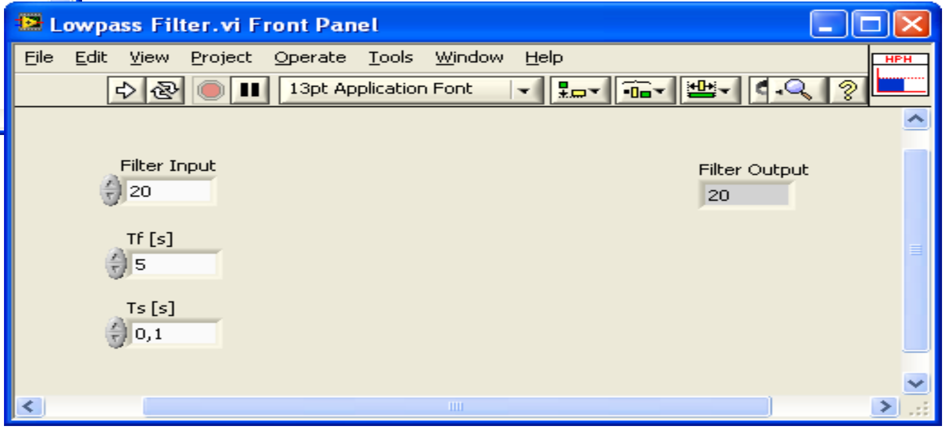
Noisy input signal (u_k)

$$T_s \leq \frac{T_f}{5}$$

This algorithm can be easly implemented in a Programming language

Low-pass Filter/Measurement Filter - Example

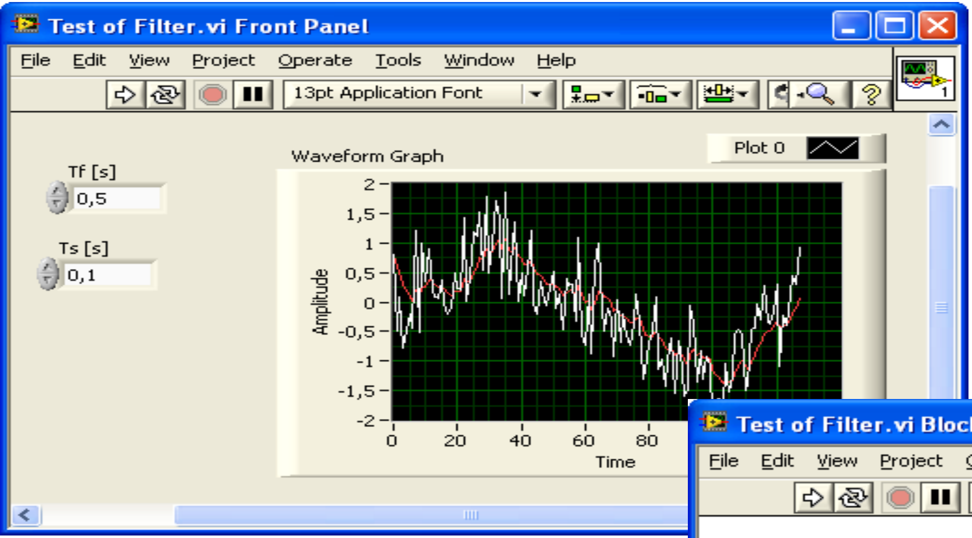




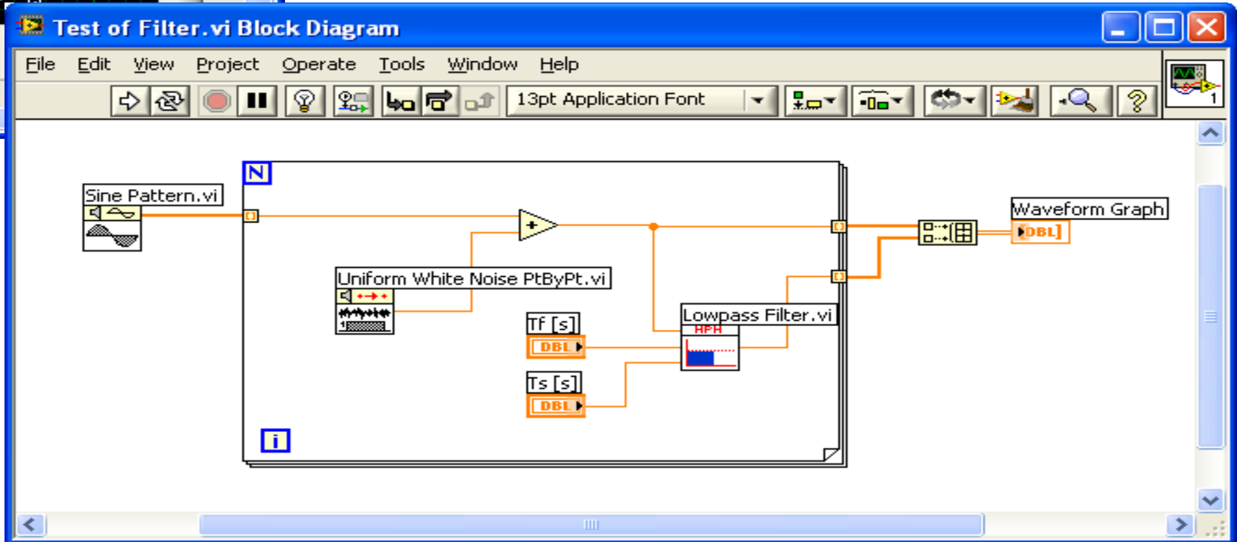
$$a = \frac{T_s}{T_f + T_s}$$

$$y_{mf}(t_k) = (1 - a)y_{mf}(t_{k-1}) + a y_m(t_k)$$

Testing the Filter

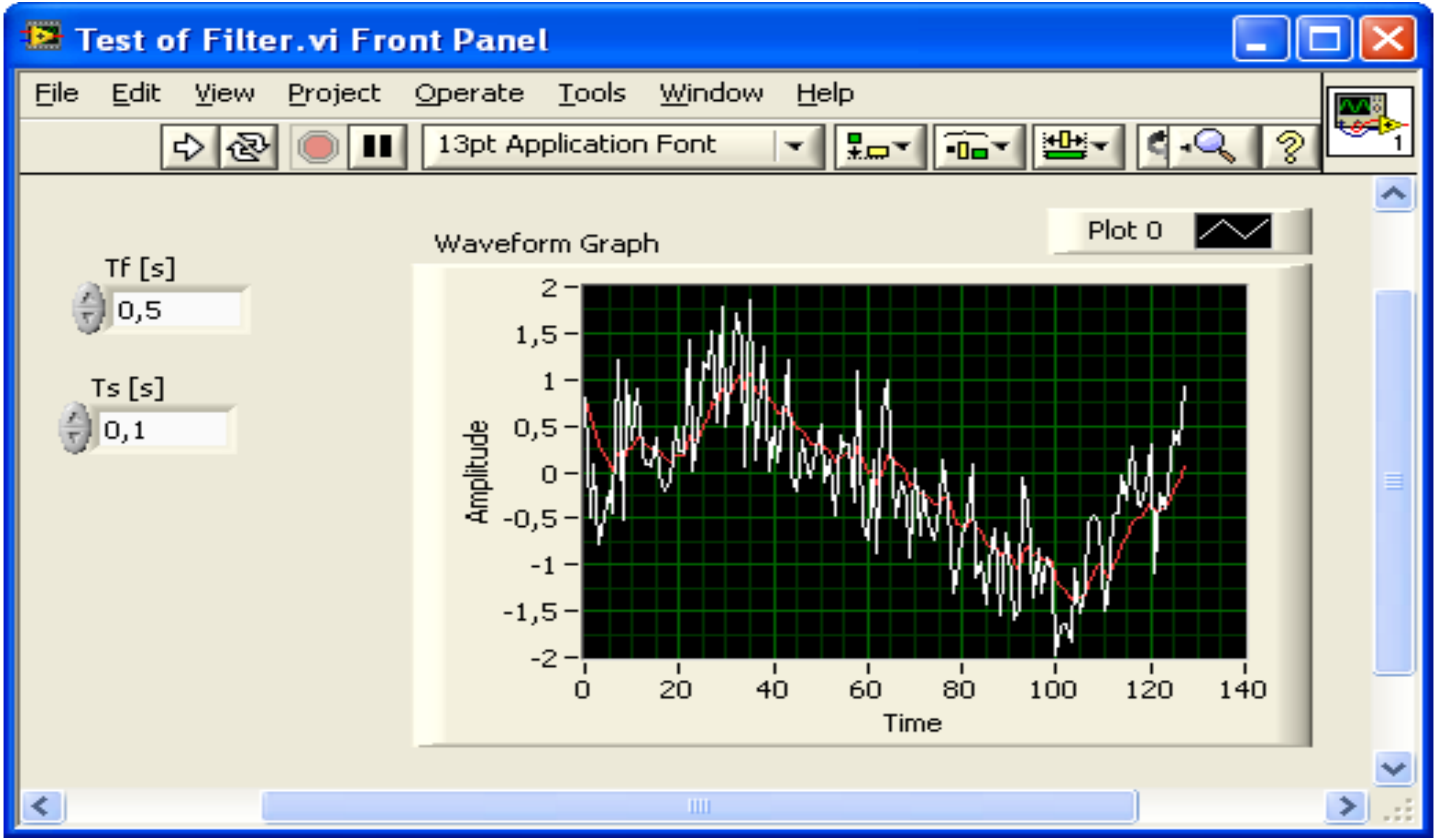


In this example we add noise to a Sine function. We then use the Measurement Filter to see if we can remove the noise afterwards.

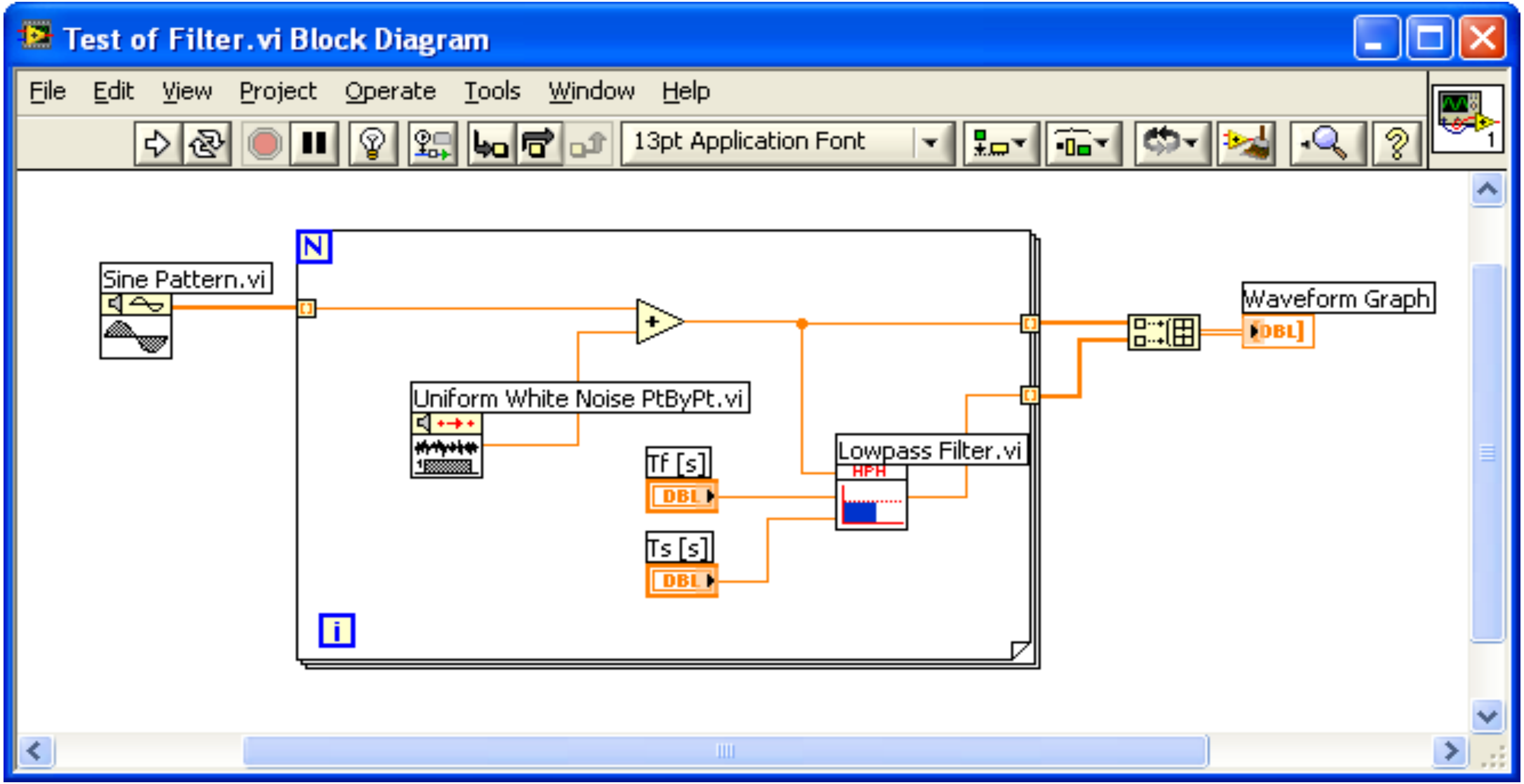


As you can see this gives good results.

The filter removes the noise from the signal.



Test of Filter.vi Front Panel
File Edit View Project Operate Tools Window Help
13pt Application Font
Tf [s]
0,5
Ts [s]
0,1
Waveform Graph
Plot 0
Amplitude
2
1,5
1
0,5
0
-0,5
-1
-1,5
-2
0 20 40 60 80 100 120 140
Time


Test of Filter.vi Block Diagram
File
Edit
View
Project
Operate
Tools
Window
Help
13pt Application Font
Sine Pattern.vi
N
Uniform White Noise PtByPt.vi
Tf [s]
DBL
Ts [s]
DBL
Lowpass Filter.vi
HPH
i
Waveform Graph
DBL

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog