PRODSP TECHNOLOGIES

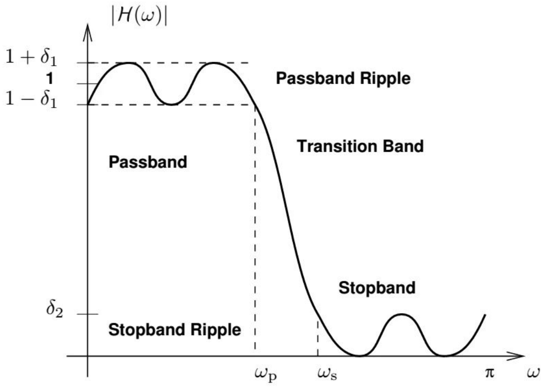
# Noise Filtering with basic Labview Functions

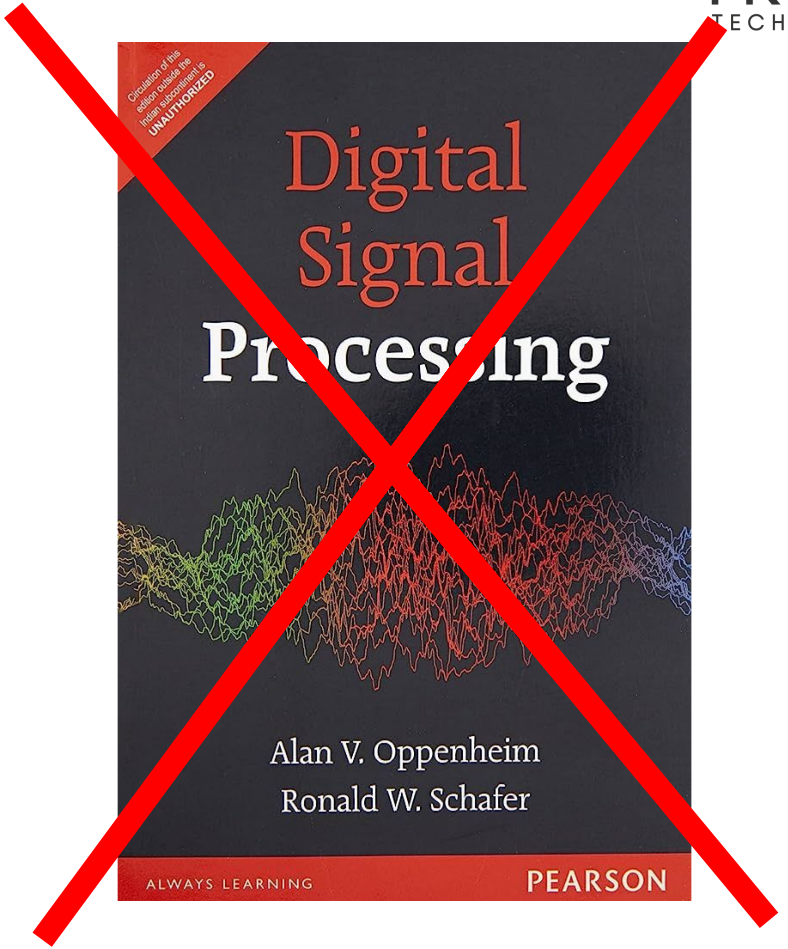
László Balogh laszlo.balogh@prodsp.hu

BudLUG Reboot, 2023-06-15

# Content

- Aim
- Manual Coding
  - Moving Average
  - Exponential Filter
- Built-in Features
- Order Estimation
- Multichannel Filtering





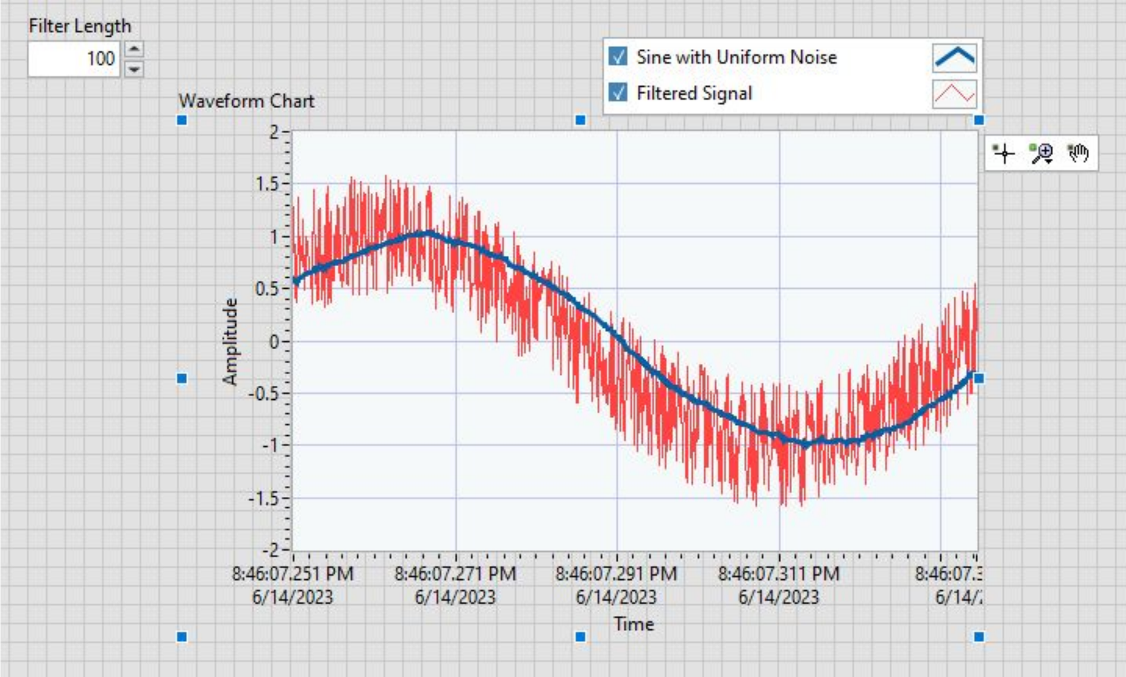
Circulation of this edition outside the Indian subcontinent is UNAUTHORIZED
Digital Signal Processing
Alan V. Oppenheim
Ronald W. Schafer
ALWAYS LEARNING
PEARSON

# Aim

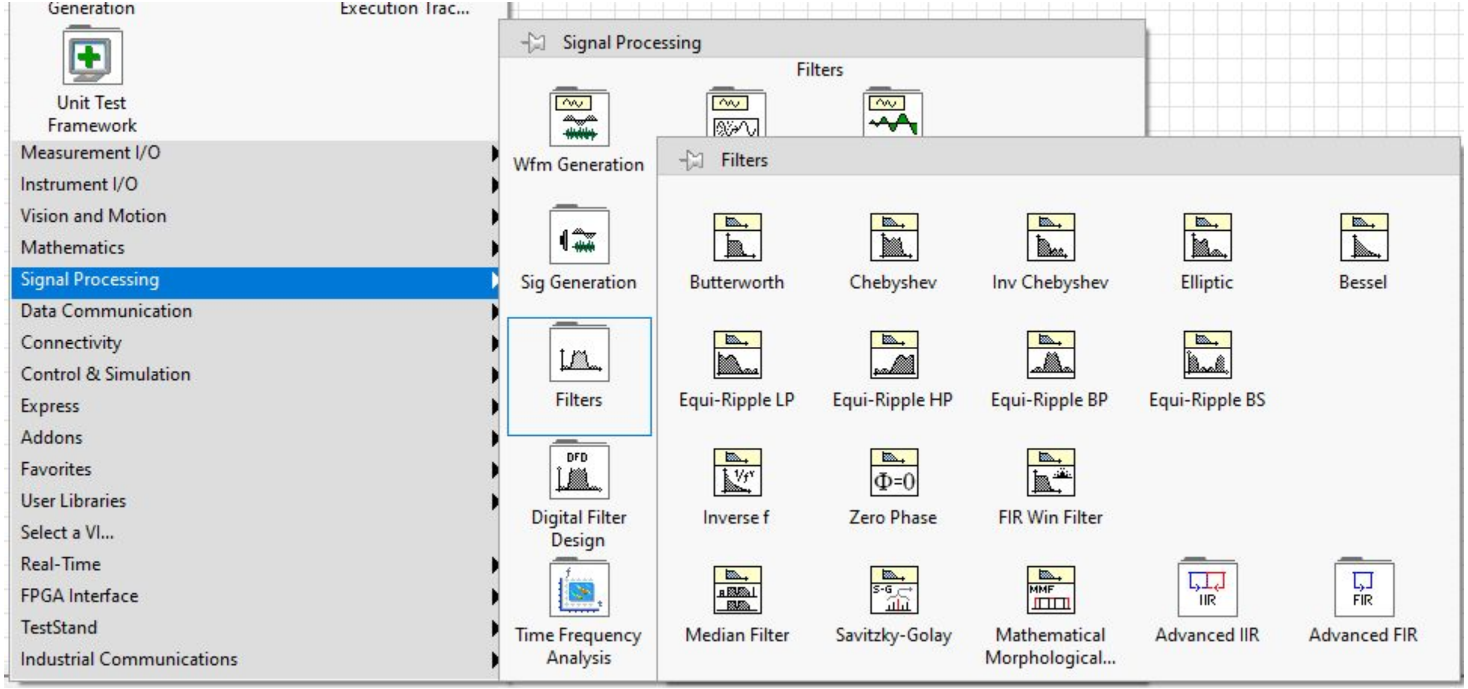
- Noise filtering for DAQ

Assumptions

- only PC platform
- 1 or more channel
- time domain requirements



# Toolset


Generation
Execution Trac...
Unit Test Framework
Measurement I/O
Instrument I/O
Vision and Motion
Mathematics
Signal Processing
Data Communication
Connectivity
Control & Simulation
Express
Addons
Favorites
User Libraries
Select a VI...
Real-Time
FPGA Interface
TestStand
Industrial Communications
Signal Processing
Filters
Wfm Generation
Sig Generation
Filters
Digital Filter Design
Time Frequency Analysis
Filters
Butterworth
Chebyshev
Inv Chebyshev
Elliptic
Bessel
Equi-Ripple LP
Equi-Ripple HP
Equi-Ripple BP
Equi-Ripple BS
Inverse f
Zero Phase
FIR Win Filter
Median Filter
Savitzky-Golay
Mathematical Morphological...
Advanced IIR
Advanced FIR

# Moving Average

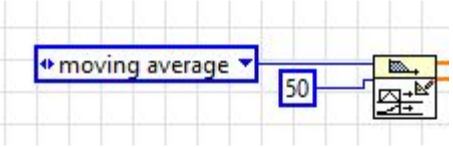
- Best theoretical description
- Straightforward implementations

$$SMA = \frac{A_1 + A_2 + \ldots + A_n}{n}$$

**where:**

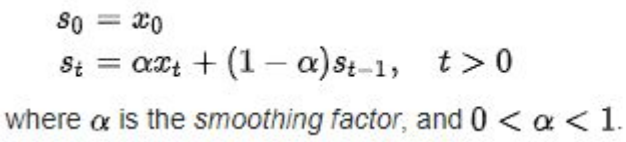
$A =$ Average in period $n$

$n =$ Number of time periods



# Exponential Filter

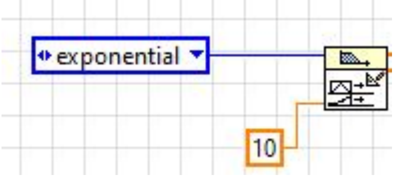
- Extremely fast



$$s_0 = x_0$$

$$s_t = \alpha x_t + (1-\alpha)s_{t-1}, \quad t > 0$$

where $\alpha$ is the *smoothing factor*, and $0 < \alpha < 1$.



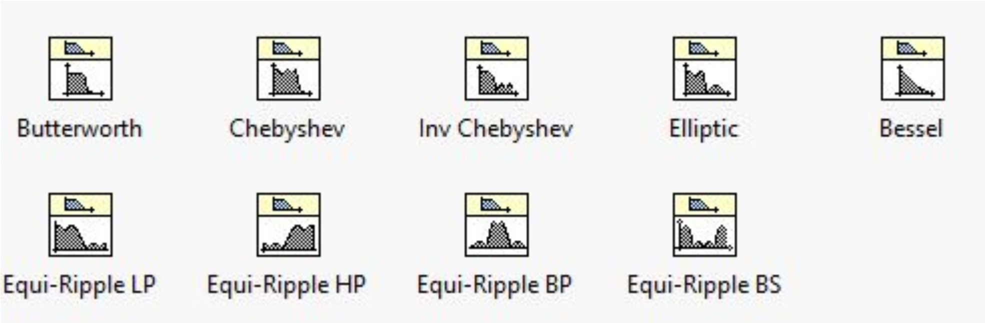
# Which One?

The **Butterworth filter** is a type of signal processing filter designed to have a frequency response that is as flat as possible in the passband.

**Chebyshev filter**s are analog or digital filters that have a steeper roll-off than Butterworth filters, and have either passband ripple (type I) or stopband ripple (type II).

An **elliptic filter** (also known as a Cauer filter) is a signal processing filter with equalized ripple (equiripple) behavior in both the passband and the stopband.

**Bessel filter** is a type of analog linear filter with a maximally flat group/phase delay (maximally linear phase response), which preserves the wave shape of filtered signals in the passband.

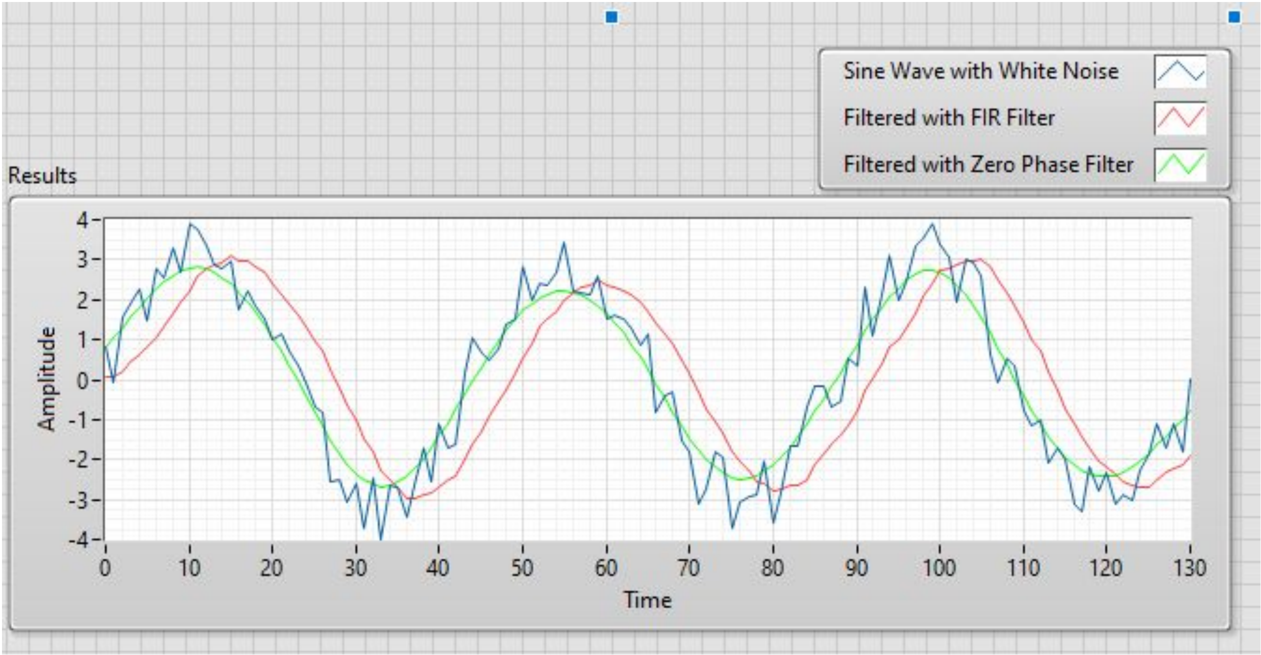


# Special Filter - Zero Phase

Φ=0

Zero Phase

- only for offline



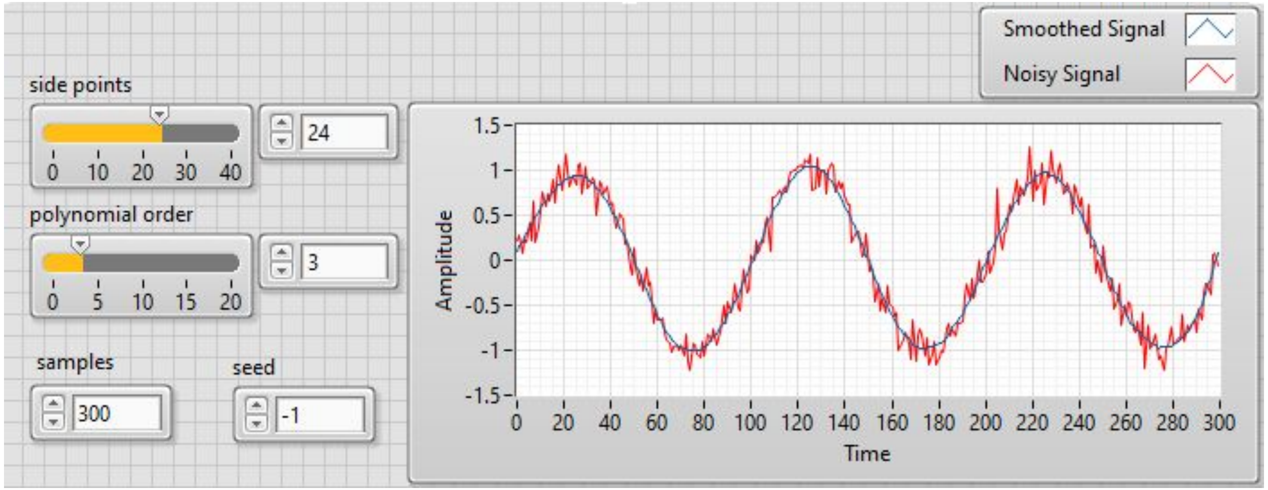
# Special Filter - Savitzky Golay

Savitzky-Golay

- convolution
- polynomial fitting (LS)

Easy to calculate

- Derivative
- Integral



side points 24

polynomial order 3

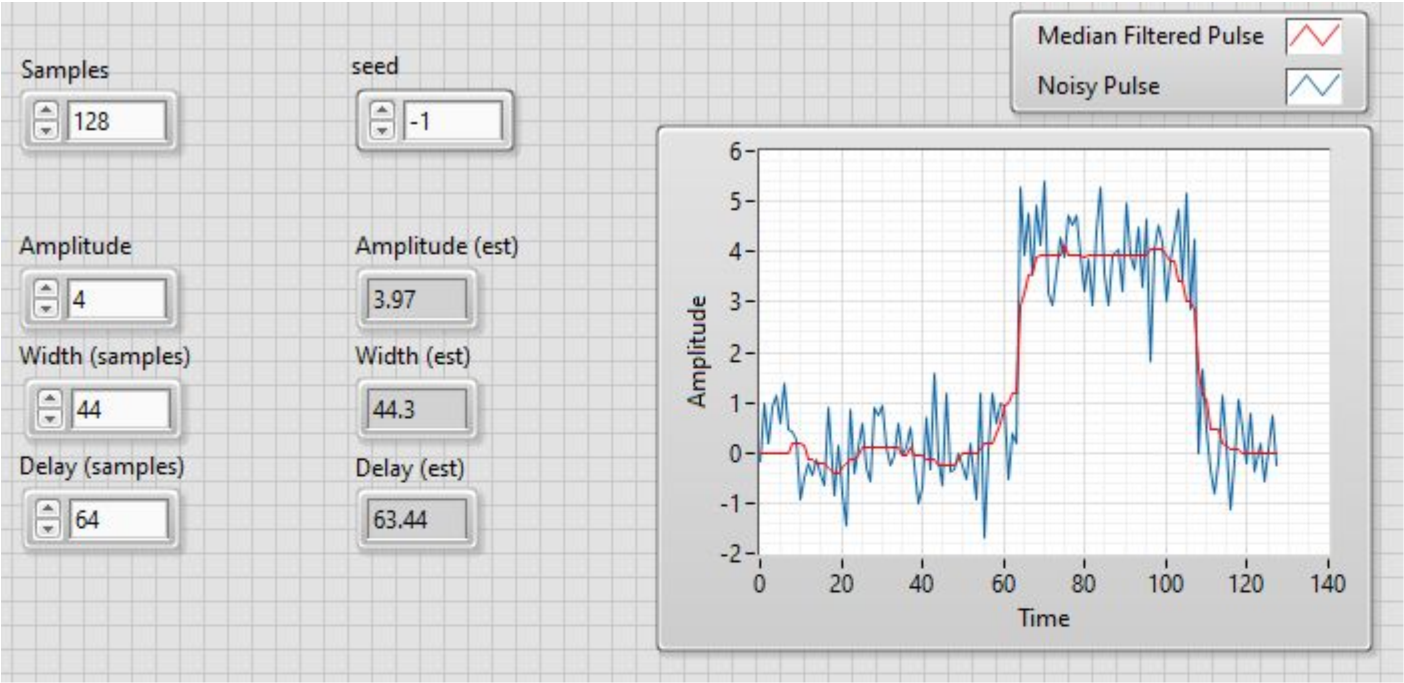
samples 300

seed -1

# Median Filter

Median Filter

- Non-linear



Samples: 128

seed: -1

Amplitude: 4

Width (samples): 44

Delay (samples): 64

Amplitude (est): 3.97

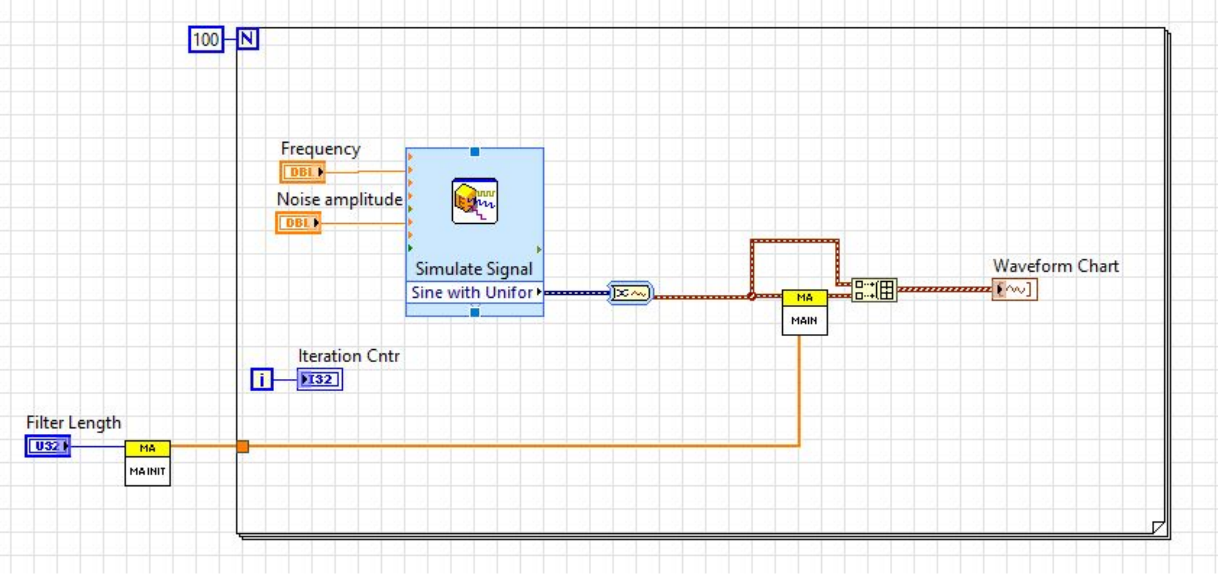
Width (est): 44.3

Delay (est): 63.44

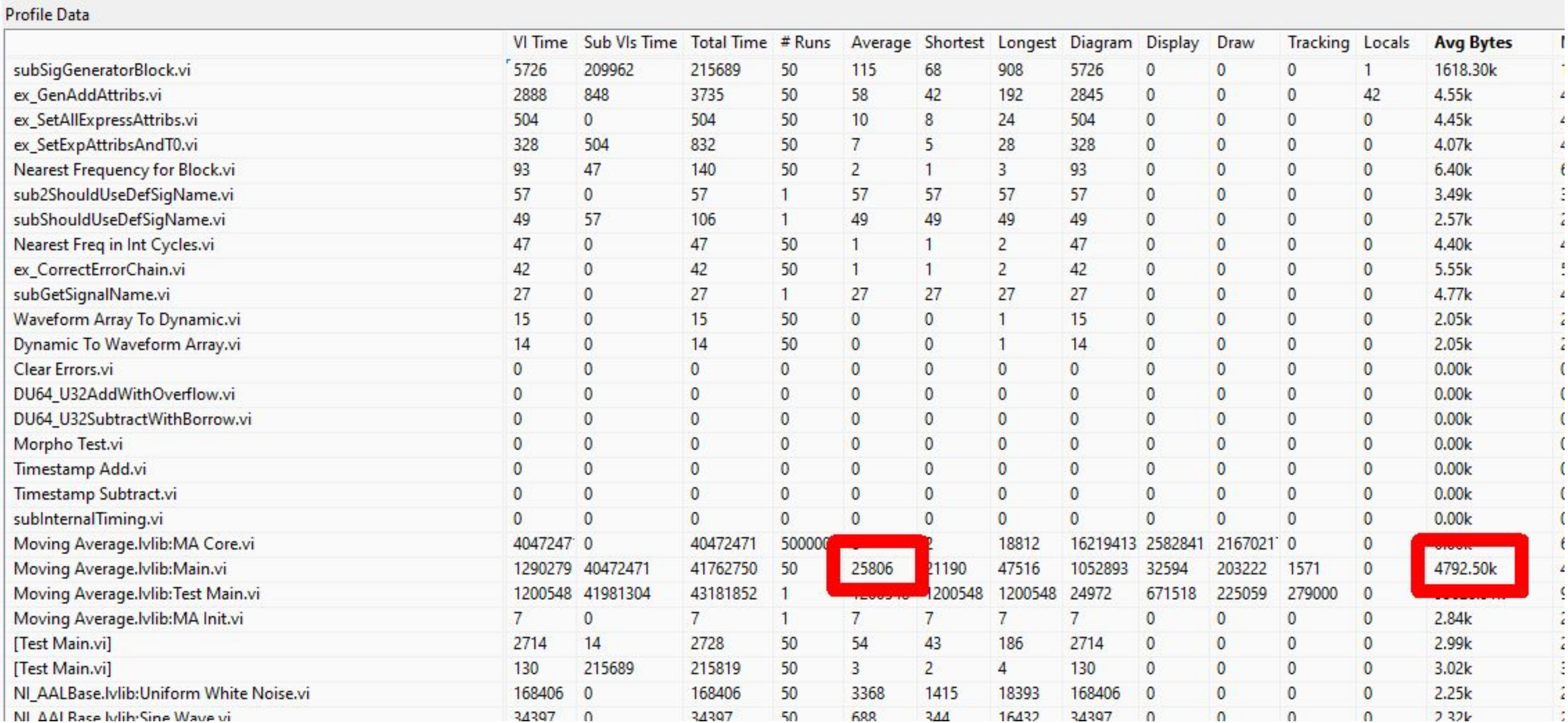
# Testing

## Key Performance Indicator

- Speed
- Memory Usage



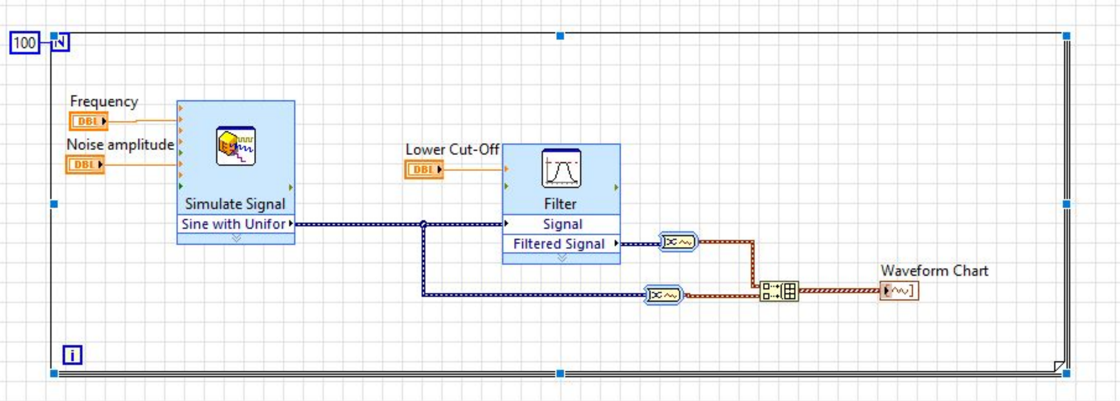
# Moving Average - Manual Implementation

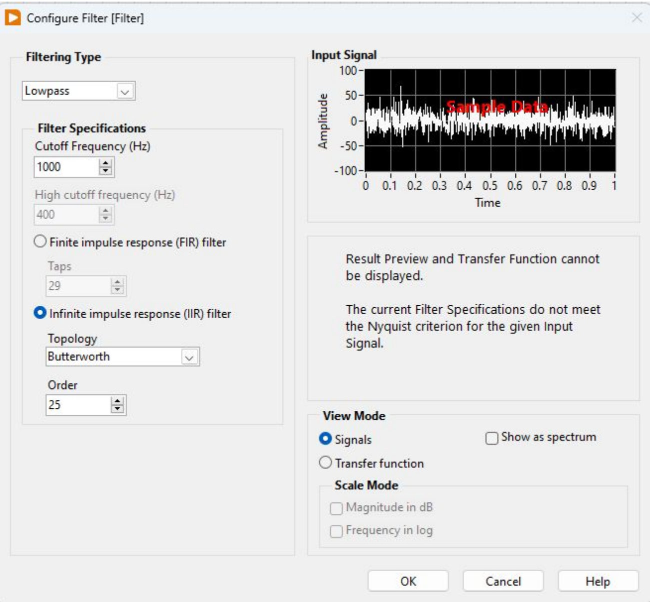


Profile Data

| | VI Time | Sub VIs Time | Total Time | # Runs | Average | Shortest | Longest | Diagram | Display | Draw | Tracking | Locals | Avg Bytes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| subSigGeneratorBlock.vi | 5726 | 209962 | 215689 | 50 | 115 | 68 | 908 | 5726 | 0 | 0 | 0 | 1 | 1618.30k |
| ex_GenAddAttribs.vi | 2888 | 848 | 3735 | 50 | 58 | 42 | 192 | 2845 | 0 | 0 | 0 | 42 | 4.55k |
| ex_SetAllExpressAttribs.vi | 504 | 0 | 504 | 50 | 10 | 8 | 24 | 504 | 0 | 0 | 0 | 0 | 4.45k |
| ex_SetExpAttribsAndT0.vi | 328 | 504 | 832 | 50 | 7 | 5 | 28 | 328 | 0 | 0 | 0 | 0 | 4.07k |
| Nearest Frequency for Block.vi | 93 | 47 | 140 | 50 | 2 | 1 | 3 | 93 | 0 | 0 | 0 | 0 | 6.40k |
| sub2ShouldUseDefSigName.vi | 57 | 0 | 57 | 1 | 57 | 57 | 57 | 57 | 0 | 0 | 0 | 0 | 3.49k |
| subShouldUseDefSigName.vi | 49 | 57 | 106 | 1 | 49 | 49 | 49 | 49 | 0 | 0 | 0 | 0 | 2.57k |
| Nearest Freq in Int Cycles.vi | 47 | 0 | 47 | 50 | 1 | 1 | 2 | 47 | 0 | 0 | 0 | 0 | 4.40k |
| ex_CorrectErrorChain.vi | 42 | 0 | 42 | 50 | 1 | 1 | 2 | 42 | 0 | 0 | 0 | 0 | 5.55k |
| subGetSignalName.vi | 27 | 0 | 27 | 1 | 27 | 27 | 27 | 27 | 0 | 0 | 0 | 0 | 4.77k |
| Waveform Array To Dynamic.vi | 15 | 0 | 15 | 50 | 0 | 0 | 1 | 15 | 0 | 0 | 0 | 0 | 2.05k |
| Dynamic To Waveform Array.vi | 14 | 0 | 14 | 50 | 0 | 0 | 1 | 14 | 0 | 0 | 0 | 0 | 2.05k |
| Clear Errors.vi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00k |
| DU64_U32AddWithOverflow.vi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00k |
| DU64_U32SubtractWithBorrow.vi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00k |
| Morpho Test.vi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00k |
| Timestamp Add.vi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00k |
| Timestamp Subtract.vi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00k |
| subInternalTiming.vi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00k |
| Moving Average.lvlib:MA Core.vi | 4047247 | 0 | 40472471 | 500000 | [illegible] | [illegible] | 18812 | 16219413 | 2582841 | 2167021 | 0 | 0 | [illegible] |
| Moving Average.lvlib:Main.vi | 1290279 | 40472471 | 41762750 | 50 | 25806 | 21190 | 47516 | 1052893 | 32594 | 203222 | 1571 | 0 | 4792.50k |
| Moving Average.lvlib:Test Main.vi | 1200548 | 41981304 | 43181852 | 1 | [illegible] | 1200548 | 1200548 | 24972 | 671518 | 225059 | 279000 | 0 | [illegible] |
| Moving Average.lvlib:MA Init.vi | 7 | 0 | 7 | 1 | 7 | 7 | 7 | 7 | 0 | 0 | 0 | 0 | 2.84k |
| [Test Main.vi] | 2714 | 14 | 2728 | 50 | 54 | 43 | 186 | 2714 | 0 | 0 | 0 | 0 | 2.99k |
| [Test Main.vi] | 130 | 215689 | 215819 | 50 | 3 | 2 | 4 | 130 | 0 | 0 | 0 | 0 | 3.02k |
| NI_AALBase.lvlib:Uniform White Noise.vi | 168406 | 0 | 168406 | 50 | 3368 | 1415 | 18393 | 168406 | 0 | 0 | 0 | 0 | 2.25k |
| NI_AALBase.lvlib:Sine Wave.vi | 34397 | 0 | 34397 | 50 | 688 | 344 | 16432 | 34397 | 0 | 0 | 0 | 0 | 2.32k |

# Express VI




Configure Filter [Filter]

**Filtering Type**

Lowpass

**Filter Specifications**

Cutoff Frequency (Hz): 1000

High cutoff frequency (Hz): 400

○ Finite impulse response (FIR) filter

Taps: 29

● Infinite impulse response (IIR) filter

Topology: Butterworth

Order: 25

**Input Signal**

Result Preview and Transfer Function cannot be displayed.

The current Filter Specifications do not meet the Nyquist criterion for the given Input Signal.

**View Mode**

● Signals　□ Show as spectrum

○ Transfer function

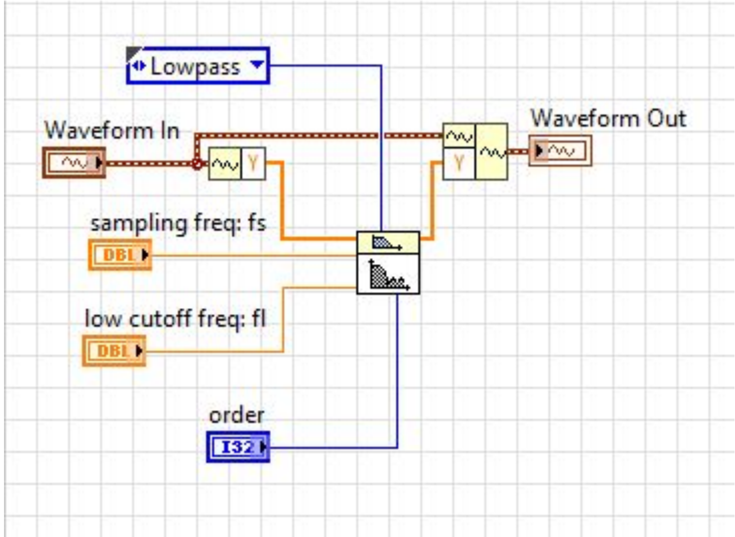
**Scale Mode**

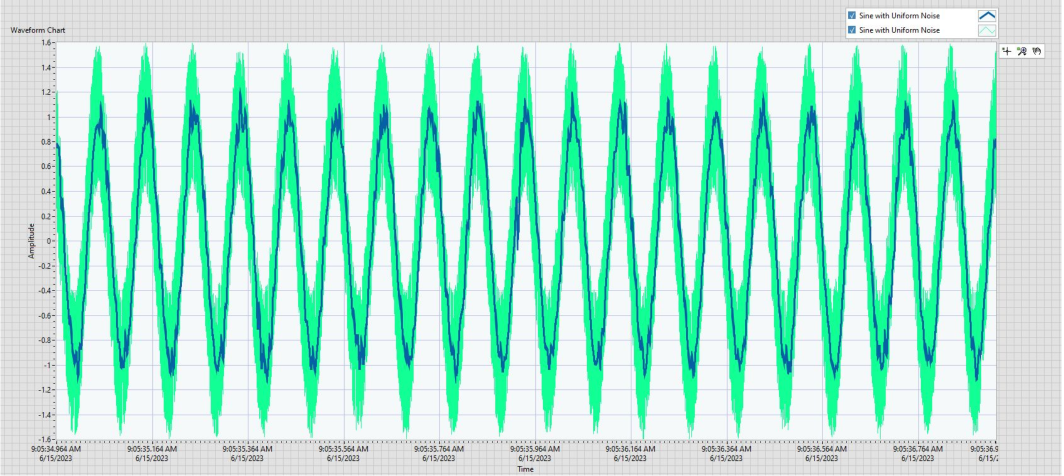
□ Magnitude in dB

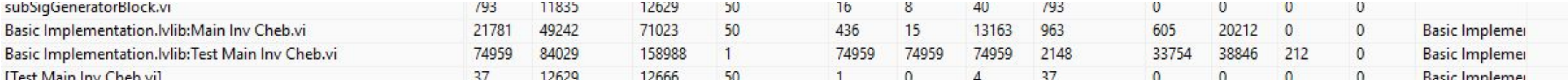
□ Frequency in log

- Performance OK
- Cannot tune filter in run-time

# Basic Implementation



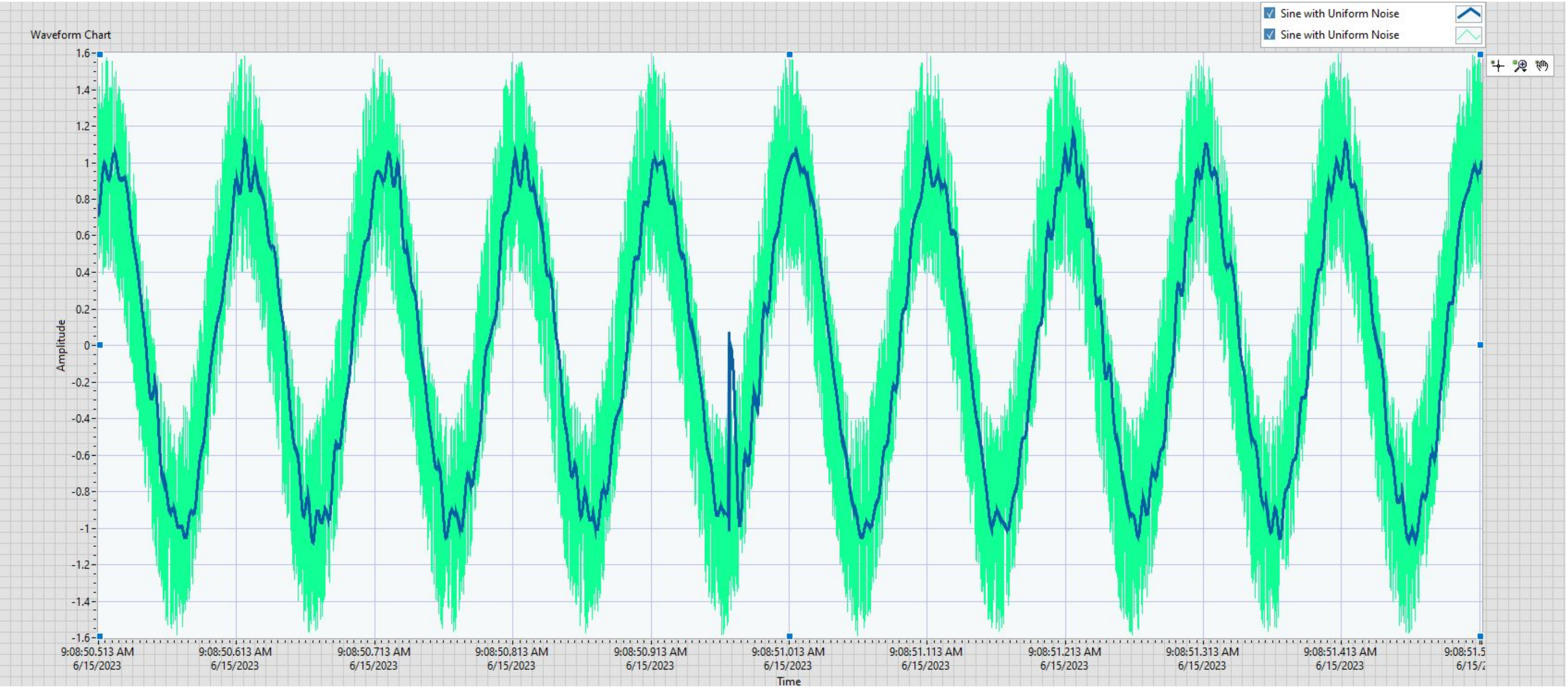




| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| subSigGeneratorBlock.vi | 793 | 11835 | 12629 | 50 | 16 | 8 | 40 | 793 | 0 | 0 | 0 | 0 | |
| Basic Implementation.lvlib:Main Inv Cheb.vi | 21781 | 49242 | 71023 | 50 | 436 | 15 | 13163 | 963 | 605 | 20212 | 0 | 0 | Basic Impleme |
| Basic Implementation.lvlib:Test Main Inv Cheb.vi | 74959 | 84029 | 158988 | 1 | 74959 | 74959 | 74959 | 2148 | 33754 | 38846 | 212 | 0 | Basic Impleme |
| [Test Main Inv Cheb.vi] | 37 | 12629 | 12666 | 50 | 1 | 0 | 4 | 37 | 0 | 0 | 0 | 0 | Basic Impleme |

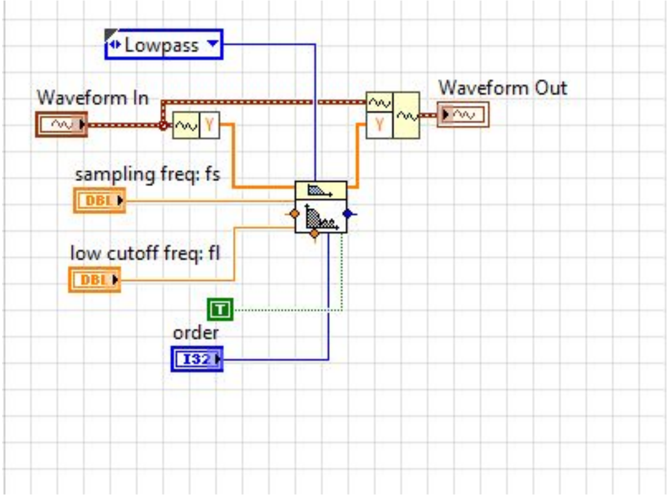
- Fast: 436 us / 1000 samples
- Small Memory Footprint: 326 kByte

# Basic Implementation - Problem

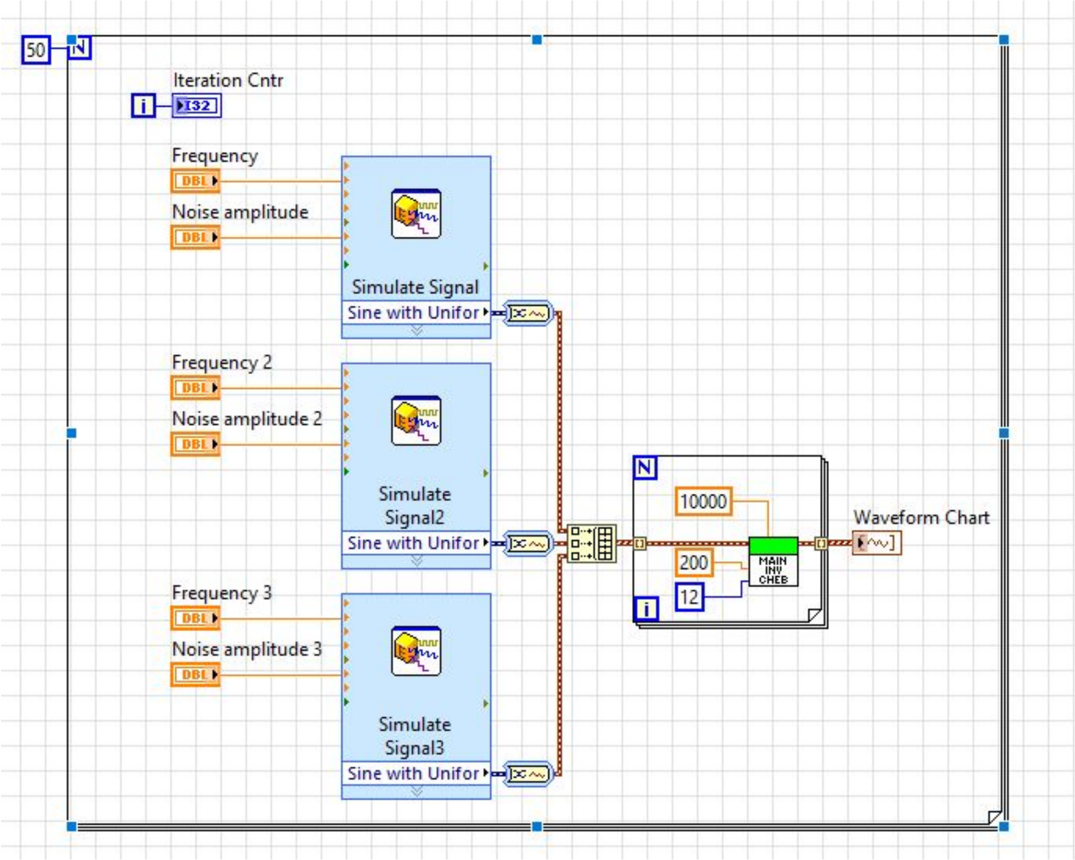


# Basic Implementation - Solution

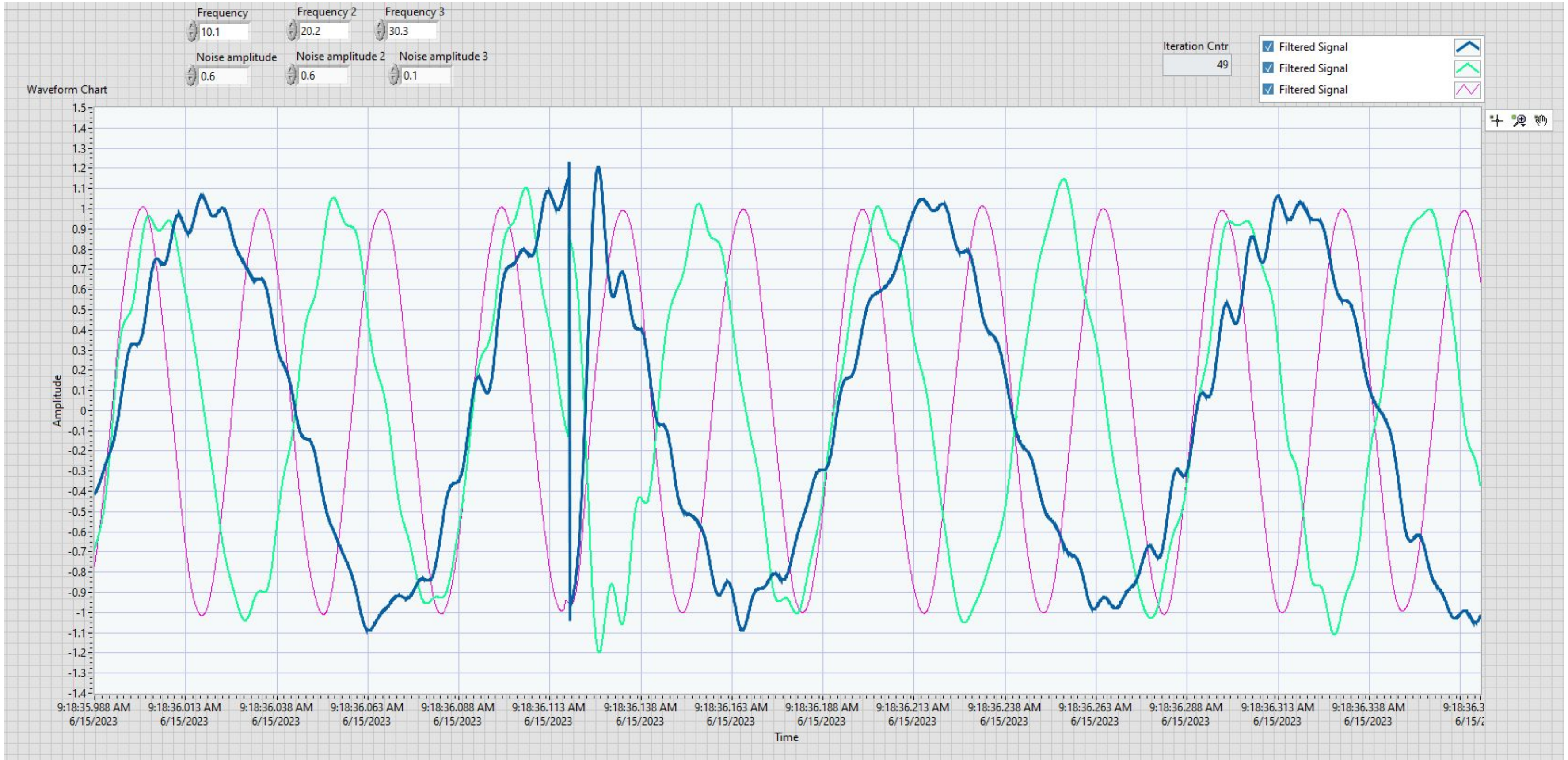
init/cont (init: F)



# Multiple Signals



# Multiple Signals - Problem



Frequency: 10.1

Frequency 2: 20.2

Frequency 3: 30.3

Noise amplitude: 0.6

Noise amplitude 2: 0.6

Noise amplitude 3: 0.1

Iteration Cntr: 49

☑ Filtered Signal

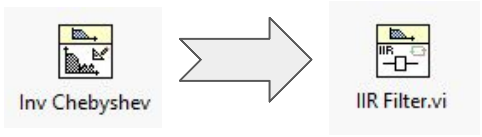
☑ Filtered Signal

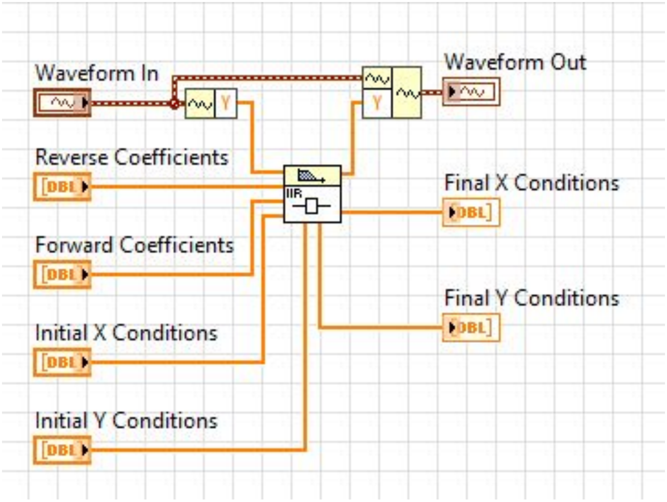
☑ Filtered Signal

# Multiple Signals - Solution

Design once

Filter N times

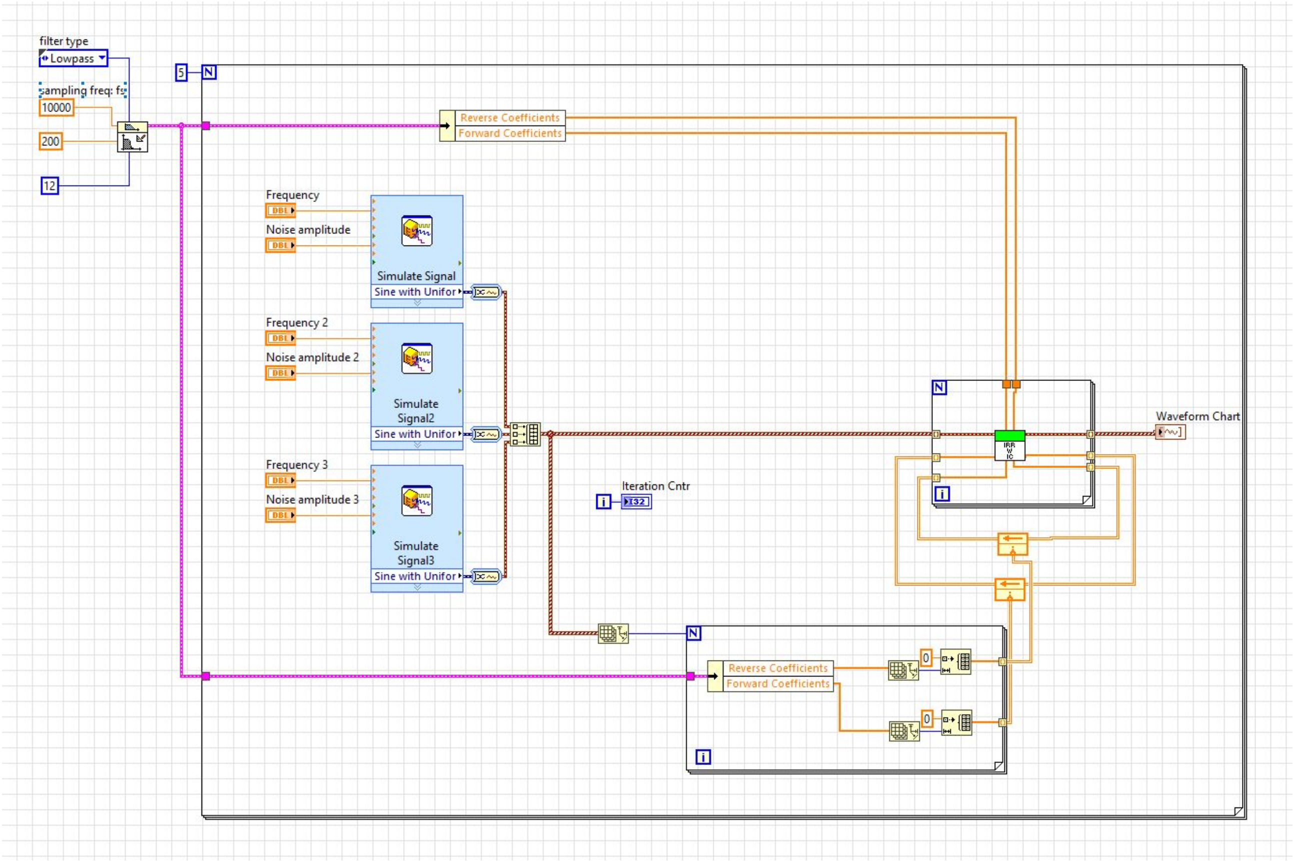




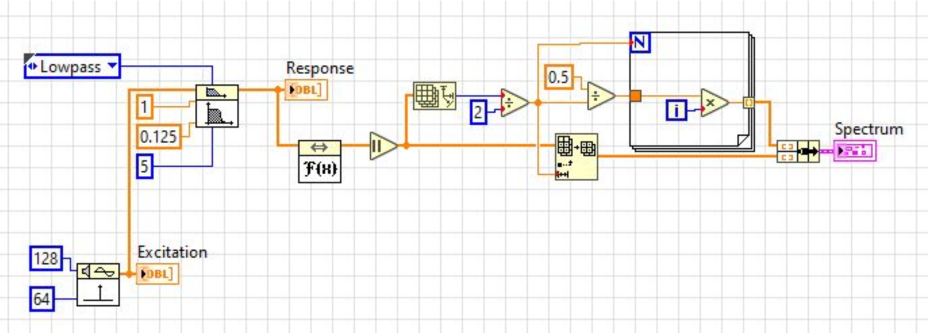
NI nomenclature:

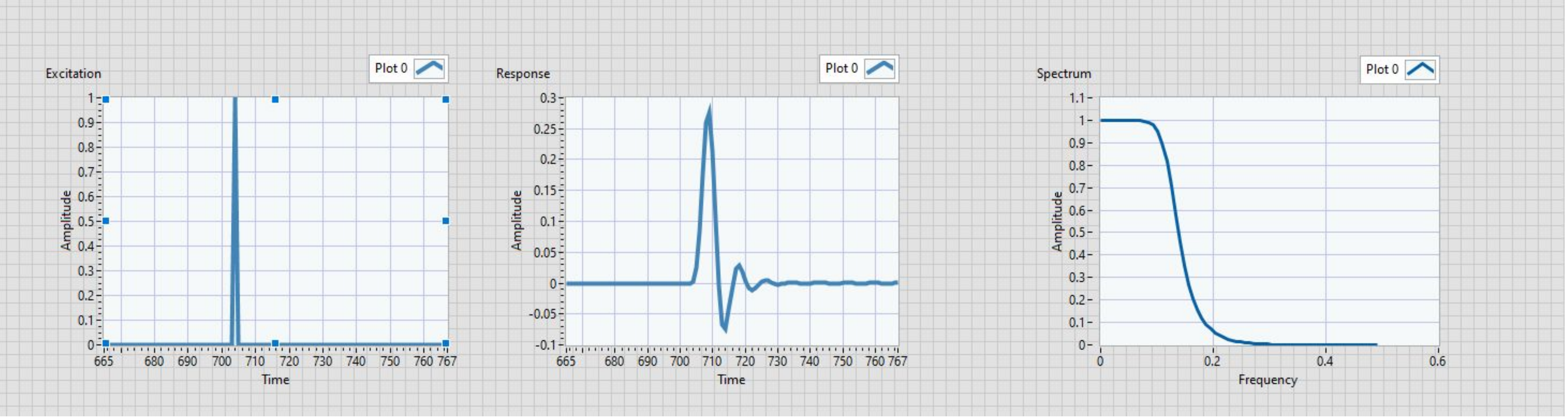
- I.C. : Initial Condition

# Multiple Signals - Solution



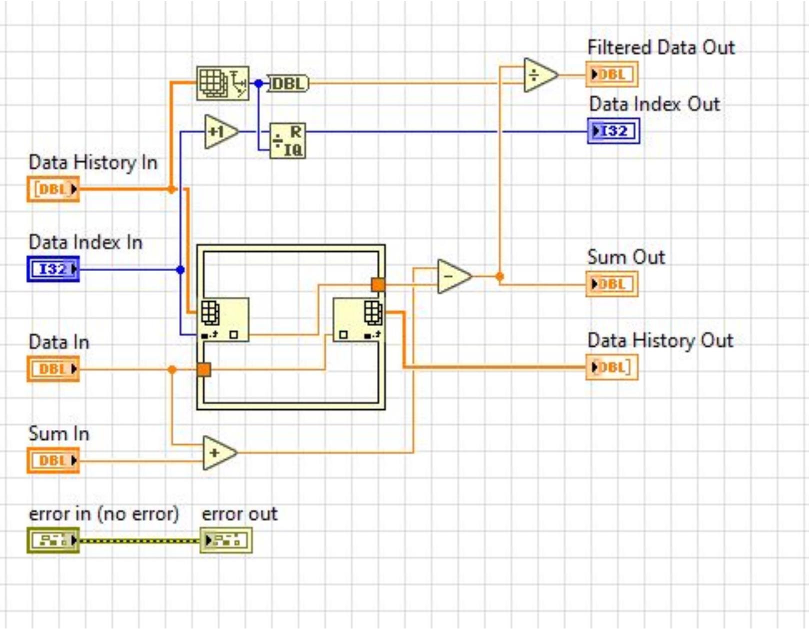
# Extra 1 - Simple Spectrum Calculation





# Extra 2 - Moving Average - Manual Implementation

- For long measurements better speed



# Conclusions

- Express VI: good solution for quick measurement

- 1 channel: broader function selection, rich API

- N channel: not all the functions are available, little bit more programming

Next steps:

- Digital Filter Design Toolkit
- Advanced Signal Processing Toolkit