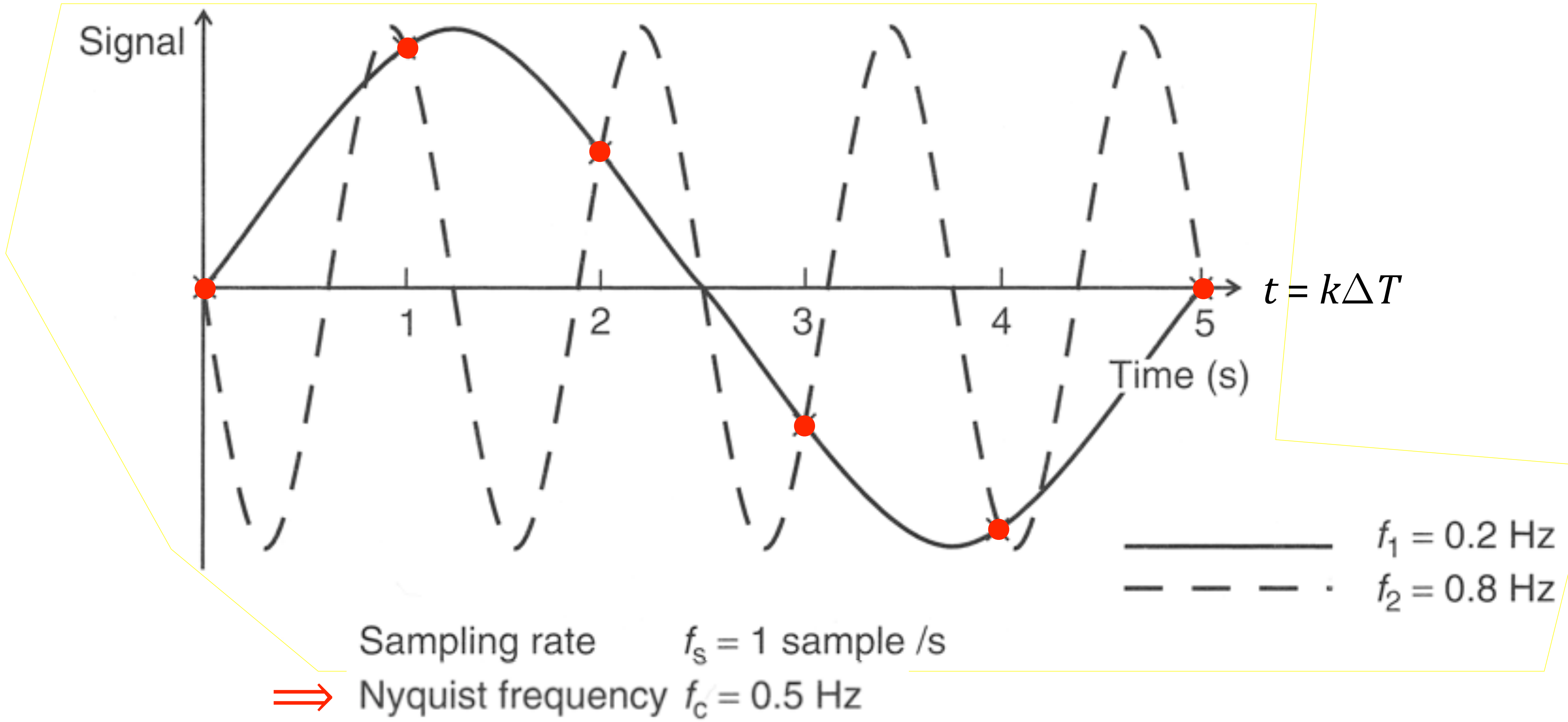
Aliasing example



The signal $\cos(2\pi f t)$ is sampled at $\Delta T = 1$ sample/s to produce the sampled signal $\cos(2\pi f k)$ (that is, $t = k\Delta T$, for $k = \ldots, -2, -1, 0, 1, 2, \ldots$). What frequencies f are aliased to $\cos(2\pi f_1 k)$ for $f_1 = 0.2$ Hz?

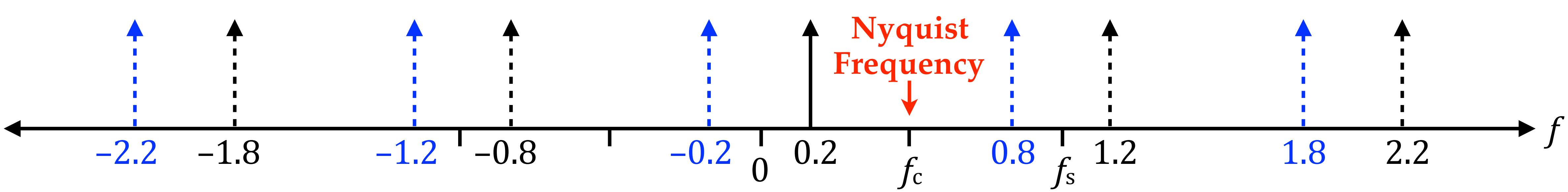
$\cos(2\pi(f_1+n)k) = \cos(2\pi f_1 k)$ for $n = \ldots, -2, -1, 0, 1, 2, \ldots$

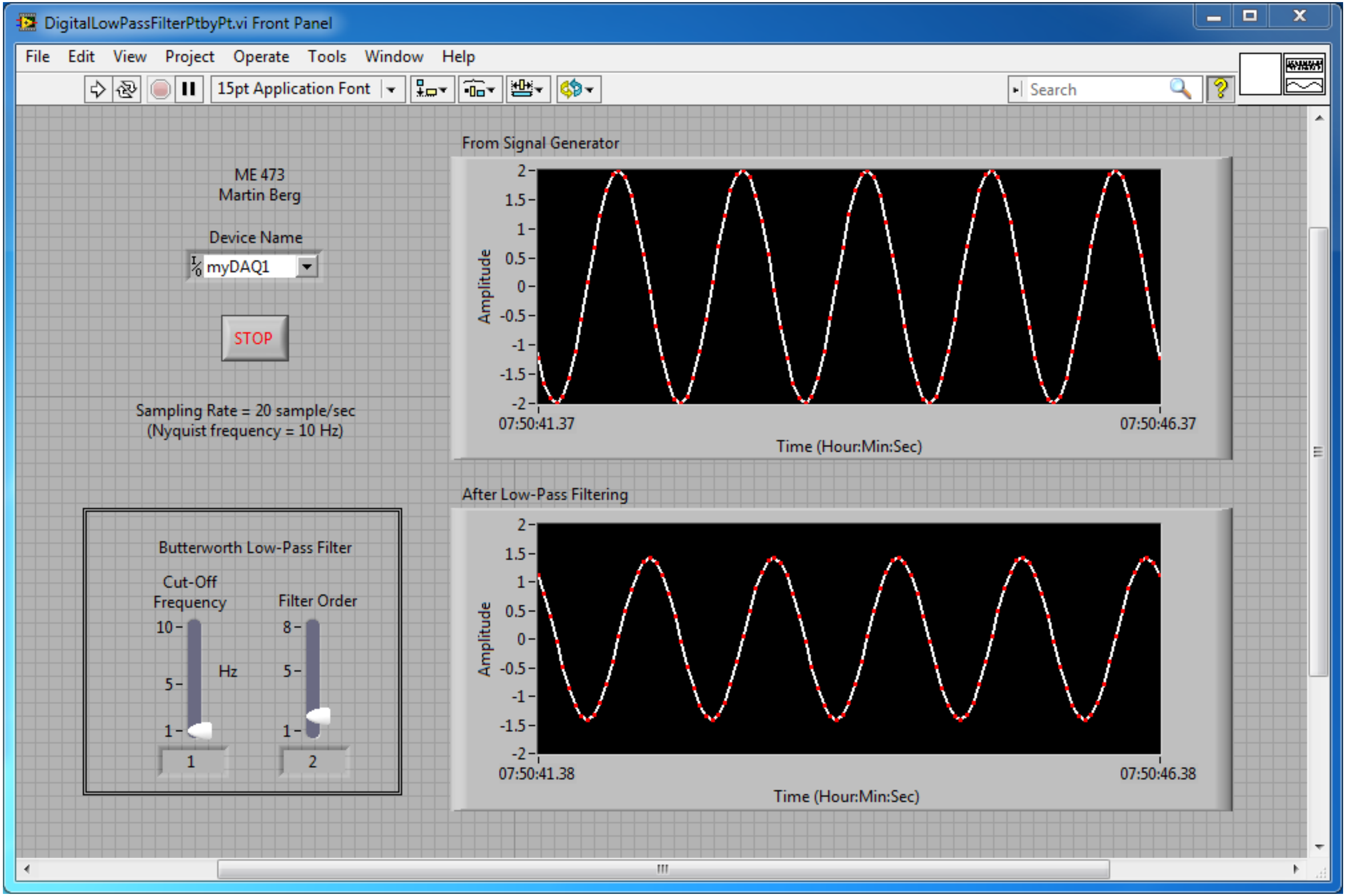
and $\cos(-2\pi(f_1+n)k) = \cos(2\pi f_1 k)$ for $n = \ldots, -2, -1, 0, 1, 2, \ldots$

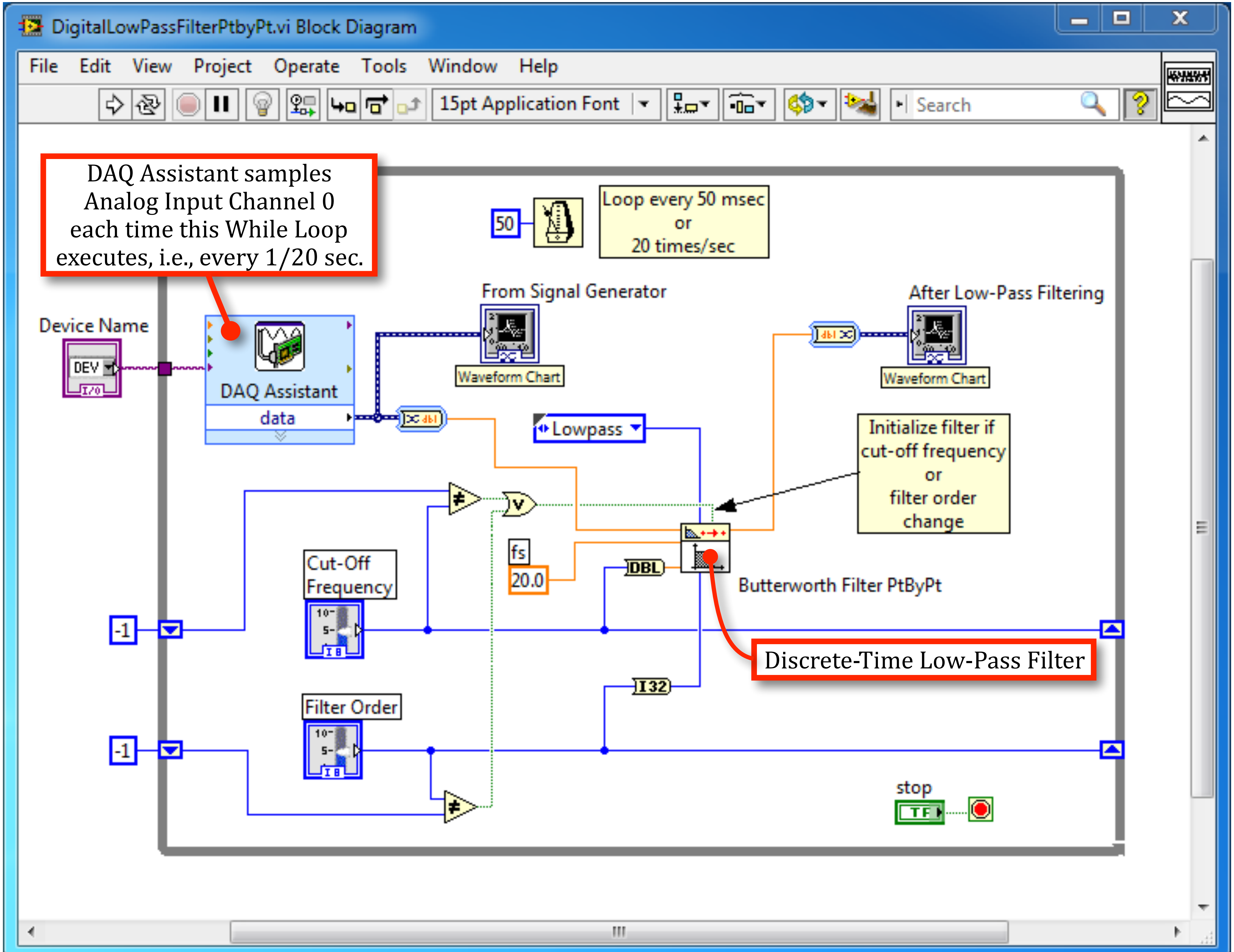
$\Longrightarrow f = \pm f_1 + n$ for $n = \ldots, -2, -1, 0, 1, 2, \ldots$

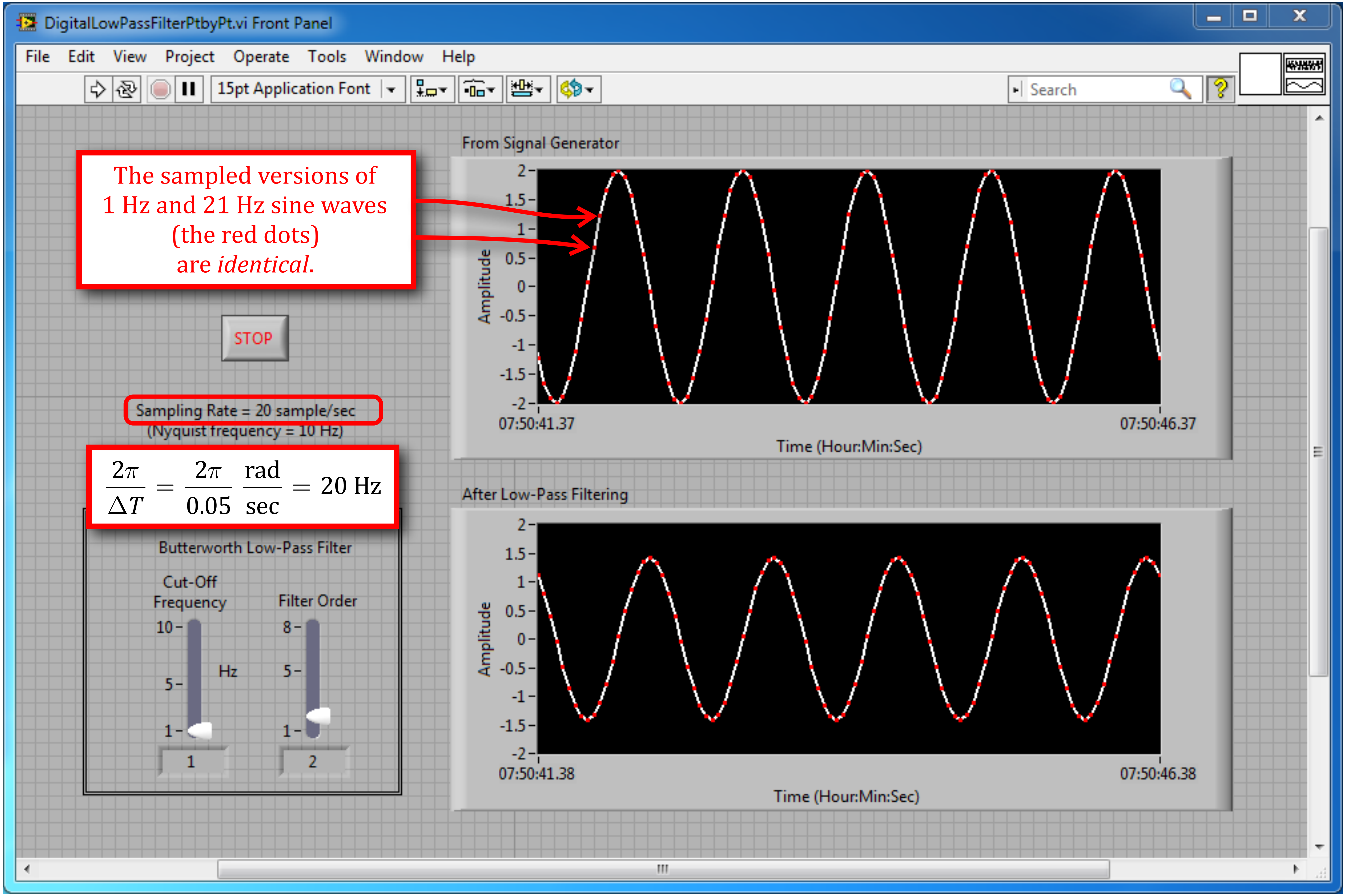
$= \pm 0.2 + n$ for $n = \ldots, -2, -1, 0, 1, 2, \ldots$

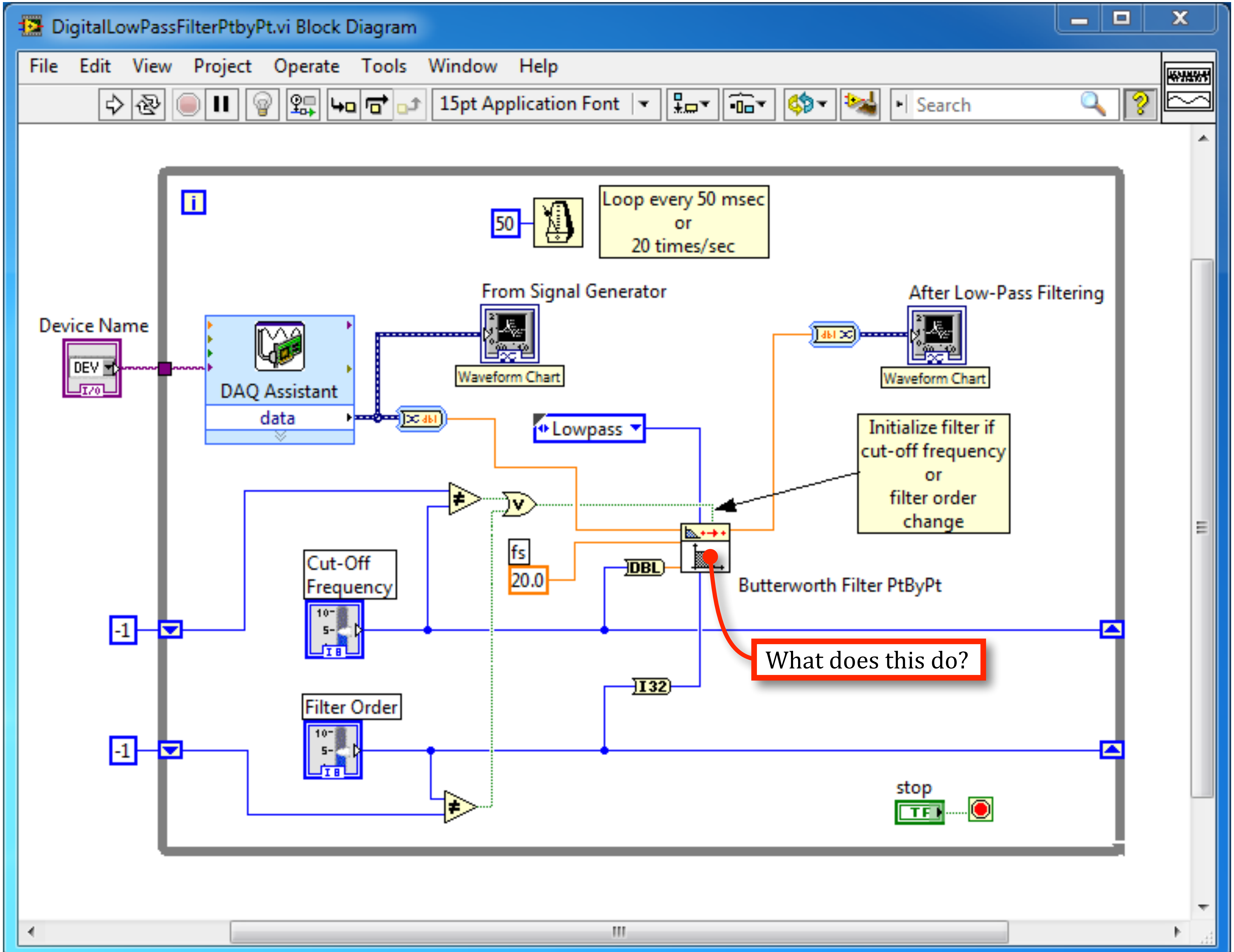
$= \pm f_1 + n/\Delta T$ in general




DigitalLowPassFilterPtbyPt.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
ME 473
Martin Berg
Device Name
myDAQ1
STOP
Sampling Rate = 20 sample/sec
(Nyquist frequency = 10 Hz)
Butterworth Low-Pass Filter
Cut-Off Frequency
Hz
Filter Order
From Signal Generator
Amplitude
07:50:41.37
07:50:46.37
Time (Hour:Min:Sec)
After Low-Pass Filtering
Amplitude
07:50:41.38
07:50:46.38
Time (Hour:Min:Sec)


DigitalLowPassFilterPtbyPt.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
DAQ Assistant samples Analog Input Channel 0 each time this While Loop executes, i.e., every 1/20 sec.
50
Loop every 50 msec or 20 times/sec
Device Name
DEV
DAQ Assistant
data
From Signal Generator
Waveform Chart
After Low-Pass Filtering
Waveform Chart
Lowpass
Initialize filter if cut-off frequency or filter order change
fs
20.0
DBL
Butterworth Filter PtByPt
Cut-Off Frequency
Filter Order
-1
-1
I32
Discrete-Time Low-Pass Filter
stop


DigitalLowPassFilterPtbyPt.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
The sampled versions of
1 Hz and 21 Hz sine waves
(the red dots)
are *identical*.
STOP
Sampling Rate = 20 sample/sec
(Nyquist frequency = 10 Hz)
$\frac{2\pi}{\Delta T} = \frac{2\pi}{0.05} \frac{\text{rad}}{\text{sec}} = 20\ \text{Hz}$
Butterworth Low-Pass Filter
Cut-Off Frequency
Hz
Filter Order
From Signal Generator
Amplitude
07:50:41.37
07:50:46.37
Time (Hour:Min:Sec)
After Low-Pass Filtering
Amplitude
07:50:41.38
07:50:46.38
Time (Hour:Min:Sec)


DigitalLowPassFilterPtbyPt.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Loop every 50 msec
or
20 times/sec
50
From Signal Generator
After Low-Pass Filtering
Device Name
DEV
DAQ Assistant
data
Waveform Chart
Waveform Chart
Lowpass
Initialize filter if
cut-off frequency
or
filter order
change
Cut-Off Frequency
fs
20.0
DBL
Butterworth Filter PtByPt
-1
-1
What does this do?
I32
Filter Order
stop

Lab 3 low-pass filter

It implements the transfer function

$$u(t) \longrightarrow \boxed{\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}} \longrightarrow y(t)$$

or, equivalently, it *solves* the corresponding differential equation

$$\frac{d^2 y}{dt^2} + 2\zeta\,\omega_n \frac{dy}{dt} + \omega_n^2\, y(t) = \omega_n^2 u(t)$$

for the output signal $y(t)$ in real time.

oard, and electrical components from the
5, the low-pass filter diagrammed below.

r circuit, choose:

$^{-6}$ F $\quad C_2 = 0.470 \times 10^{-6}$ F

ise resistances of your two resistors and

ircuit with a voltage follower.

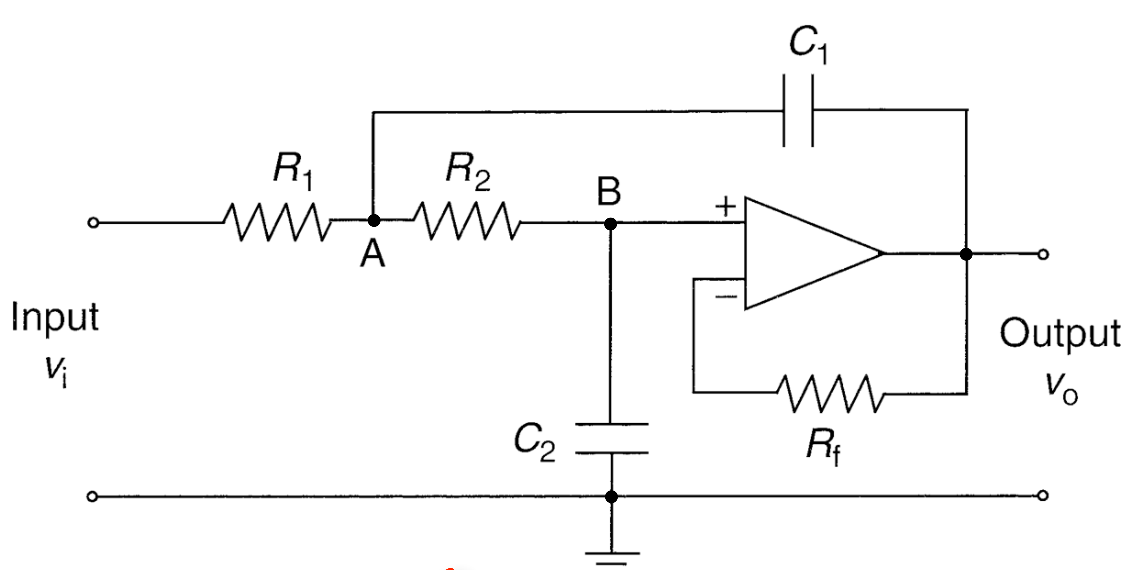
uit with another voltage follower.

$$\frac{V_o(s)}{V_i(s)} = \frac{1}{\tau_1\tau_2 s^2 + (\tau_2 + \tau_3)s + 1} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$\tau_1 = R_1 C_1 \qquad \tau_2 = R_2 C_2 \qquad \tau_3 = R_1 C_2$$

$$\omega_n = \frac{1}{\sqrt{\tau_1\tau_2}} = \frac{1}{\sqrt{R_1 C_1 R_2 C_2}}$$

$$\zeta = \frac{\tau_2 + \tau_3}{2\sqrt{\tau_1\tau_2}} = \frac{(R_1 + R_2)C_2}{2}\,\omega_n$$



What does this circuit do?

It implements a transfer function

$$u(k) \longrightarrow \boxed{\dfrac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_0}} \longrightarrow y(k)$$

$$n \geq m$$

or, equivalently, it *solves* the corresponding difference equation

$$\begin{aligned} y(k) + a_{n-1}\, y(k-1) + \cdots + a_0\, y(k-n) \\ = b_m u(k-(n-m)) + b_{m-1} u(k-(n-m)-1) + \cdots + b_0 u(k-n) \end{aligned}$$

for the output signal $y(k)$ in real time.