LAB EXERCISE #9: C++ Programming
Finite Impulse Response Filter (Array Usage)

Prelab:

Preliminary C++ Program

Introduction:

An FIR (Finite Impulse Response) filter can be represented by

$$y[n] = \sum_{k=0}^{\min(N-1,n)} x[n-k]h[k] \qquad (1)$$

where N is the number of coefficients, $h[k]$ is the impulse response of the filter, $x[n-k]$ is the input of the filter delayed k clock cycles, $y[n]$ is the output of the filter, and $x[m] = 0$ for $m < 0$.

A discrete filter accepts a sequence of data (input), one at a time, and produces a new sequence of data (output), one at a time. In the above equation, n is the time index, $n = 0, 1, 2, \ldots$. The filter behavior is explained below.

At $n = 0$, the filter gets only one input which is $x[0]$, so the filter uses $x[0]$ to generate the output $y[0]$ at time moment of $n = 0$. At $n = 1$, the filter has two inputs available which are $x[0], x[1]$, so it will use these two values to generate the output $y[1]$. When the number of input data grows higher than 35, the filter only uses the latest 35 inputs to compute the output. In other words, equation (1) can be split into the following two equations:

$$y[n] = \sum_{k=0}^{n} x[n-k]h[k], \qquad n < N \qquad (2)$$

and

$$y[n] = \sum_{k=0}^{N-1} x[n-k]h[k], \qquad n \geq N \qquad (3)$$

where $N = 35$ which is the order number for the filter we use below.

From equation (2), we can calculate $y[n]$ for $n < 35$ as follows:

$$y[0] = h[0]$$

$$y[1] = x[1]h[0] + x[0]h[1]$$

$$y[2] = x[2]h[0] + x[1]h[1] + x[0]h[2]$$

$$\ldots$$

From equation (3), we can calculate $y[n]$ for $n \geq 35$ as follows:

$$y[35] = x[35]h[0] + x[34]h[1] + \cdots x[3]h[32] + x[2]h[33] + x[1]h[34]$$
$$y[36] = x[36]h[0] + x[35]h[1] + \cdots x[4]h[32] + x[3]h[33] + x[2]h[34]$$
$$y[37] = x[37]h[0] + x[36]h[1] + \cdots x[5]h[32] + x[4]h[33] + x[3]h[34]$$
$$\cdots$$

A given optimal equiripple FIR (Finite Impulse Response) filter has the following specification:
Sample rate: 10 kHz (0.0001 sec / sample);

Passband: 0 – 2.5 kHz (lowpass), 0.5 dB maximum ripple;

Stopband: 3.0 – 5 kHz, 50dB minimum attenuation.

Using the Parks-Mcclellan program, we obtain the following impulse response values (scaled up by 2^{15}; ie. to get actual value, divide by 2^{15}), with $N = 35$.

h(0) = h(34) = 361.922	h(9) = h(25) = 812.822
h(1) = h(33) = 589.000	h(10) = h(24) = -934.419
h(2) = h(32) = 52.556	h(11) = h(23) = -1082.725
h(3) = h(31) = -538.095	h(12) = h(22) = 1547.666
h(4) = h(30) = -58.657	h(13) = h(21) = 1083.109
h(5) = h(29) = 499.472	h(14) = h(20) = -3229.928
h(6) = h(28) = -251.531	h(15) = h(19) = -1275.738
h(7) = h(27) = -785.168	h(16) = h(18) = 10268.660
h(8) = h(26) = 381.999	h(17) = h(17) = 17571.900

Courtesy: Dale Clover & John Deller, Digital Signal Processing and Microcontroller, Motorola University, Prentice-Hall, 1999.

Tasks:

Write a C++ program to do steps 1 to 4 and use Excel to do step 5:

1. Create three arrays $x[600], h[35]$, and $y[600]$.
2. Fill the x array with the 600 sample values of $x(t) = 100\sin(4000\,\pi t)$ for $0 \leq t < 0.06$ with rate of $\Delta t = 0.0001$ per sample. Note that the frequency for the sinewave is 2 KHz.
3. Compute $y[n]$ for $0 \leq t < 0.06$.
4. Store the 600 sample values of $x[n]$ in file **filex** and store the 600 sample values of $y[n]$ in file **filey**.
5. Plot $x[n]$ and $y[n]$ versus n on the same graph using Excel. Compare and comment on the amplitude of the two sinewaves $x[n]$ and $y[n]$.

Program Outcomes and Discussions:

1. Repeat steps 1 to 5 for $x(t) = 100\sin(4000\,\pi t)$, frequency 2 kHz.
2. Repeat steps 1 to 5 for $x(t) = 100\sin(6000\,\pi t)$, frequency 3 kHz.
3. Repeat steps 1 to 5 for $x(t) = 100\sin(8000\,\pi t)$, frequency 4 kHz.
4. Compare the amplitudes of the two sinewaves $x[n]$ and $y[n]$ for frequencies 2 kHz, 3 kHz, and 4 kHz. Categorize the filter either as low pass filter, high pass filter, or band-pass filter.

Show Instructor or Lab Assistant:

Show well-documented and correct C++ program, and run output to verity that your program work as intended.