Jonathan W. Valvano
April 5, 2013, 10:00 to 10:50am

(10) Question 1. The important concept in this problem is that real-time specification depends on the frequency of the signal. As the frequency increases the time jitter specification must reduce.

Step 1. Find the derivative of $V(t) = A+B\mathbf{cos}(2\pi ft)$

$dV(t)/dt = -2\pi f B\, \mathbf{sin}(2\pi ft)$;units V/sec

Step 2. Find the maximum of the derivative

$|dV(t)/dt_{\text{max}}| = 2\pi f B$;units V/sec

Step 3. Multiply the maximum of the derivative times the time-jitter, δt,

$\delta V = 2\pi f B\, \delta t$;units V

(20) Question 2. A 10-bit ADC is sampled at 4096 Hz and converted to the frequency.

Part a) First describe what the magnitude axis means in terms of the ADC bits.

10 bits is equivalent to $20\log_{10}(1/1024) \approx -60$dB full scale
12 bits is equivalent to $20\log_{10}(1/4096) \approx -72$dB full scale

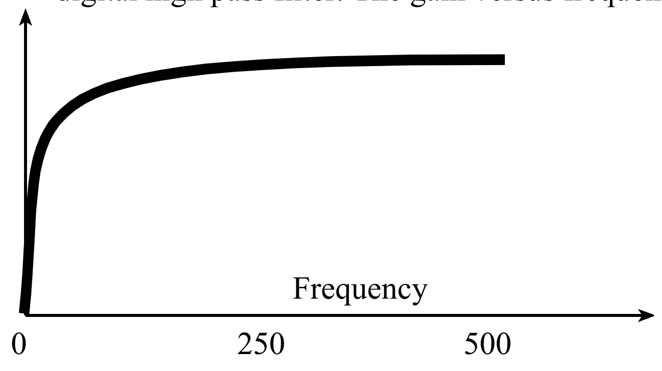
If we changed the number of ADC bits from 10 to 12, then the range of valid data would change from 0 to -60dB to 0 to -72dB.
C- The magnitude axis would change from a range 0 through -60 dB to a range of 0 to -72 dB.

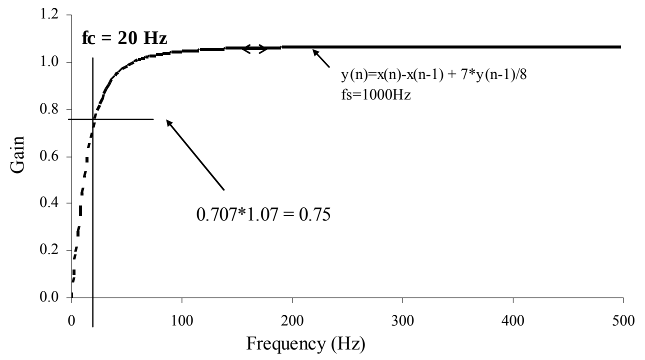
Part b) The number of points determines the frequency resolution, $\Delta f=f_s/N$. If we changed the number of points in the FFT from 4096 to 8192, the frequency resolution would change from 1 Hz to ½ Hz, meaning there would be twice as many points along the frequency axis.
I- There would be twice as many data points, placing a new value in between each of the existing.

Part c) The sampling rate determines the range of frequencies, according to Nyquist, which will be 0 to ½ f_s. If we changed the sampling rate from 4096 to 8192 Hz, the frequency range would change from 0 to 2048 Hz to a range of 0 to 4096 Hz.
G- The frequency axis would change from a range of 0 to 2048 Hz to a range of 0 to 4096 Hz.

Part d) Since there is a large 60 Hz component, the most likely cause of this noise is EM field pickup.

(30) Question 3. The sampling rate of a real-time data acquisition system is 1000 Hz.
Part a) We travel along the unit circle, $z=e^{2\pi f/fs}$, the distance to a zero makes the gain go down, and the distance to a pole makes the gain go up. Because of the zero at z=1, which is f=0, this is a digital high pass filter. The gain versus frequency plot should go from 0 to ½ f_s.





Part b) There is a zero at *z*=1 and a pole at *z*=7/8. The *H(z)* is

$$H(z)=\frac{z-1}{z-\frac{7}{8}}$$

Convert to standard form (negative exponents on *z* terms), by dividing top and bottom by *z*

$$H(z)=\frac{1-z^{-1}}{1-\frac{7}{8}z^{-1}}$$

Write in terms of *Y(z)* and *X(z)*

$$Y(z)\left(1-\tfrac{7}{8}z^{-1}\right)=X(z)\left(1-z^{-1}\right)$$

Take inverse transform to get *y(n)* in terms of *x(n)*

y(n)-7/8 y(n-1)= x(n)- x(n-1)

Write in standard format

*y(n) = x(n) – x(n-1) + (7*y(n-1))/8*

The magnitude at 500 Hz (z = -1) is 2/(1+7/8) = 16/15.

Part c) Write a C function using integer math to implement this digital filter. Multiply by 7/8 is implemented as y - y/8, the y/8 is a shift right by 3. The function prototype is

```
short Filter(short input){short output;
short static y=0;  // MACQ length 1, this is y(n-1)
short static x=0;  // MACQ length 1, this is x(n-1)
  output = input - x + y - y>>3;
  x = input;
  y = output;
  return y;
}
```

Check out the Excel file that solves this class of digital HPF for any α

$$H(z)=\frac{1-z^{-1}}{1-\alpha\cdot z^{-1}}$$ http://users.ece.utexas.edu/~valvano/EE345M/DigitalHighPassFilter.xls

(40) Question 4. The file system uses a **file table** (**FT**) to define the set of blocks allocated to file.
(5) Part a) No. At this point there is no external fragmentation because there are 65523 free blocks (65536-directory-FT-11) and we can make one file using all the available blocks. Put the new **FT** in block 1; this **FT** contains

0	3	4
1	8	8
2	11	11
3	13	21
4	27	65535

(5) Part b) Yes. Since each file has one **FT** and each **FT** fits into one block, each file has at most 64 pairs to define its blocks. In the figure the disk has 6 groups of contiguous blocks. If the disk gets in a state where it has more than 64 groups of contiguous blocks than the largest file that can be allocated will be smaller than the total number of free blocks, which is external fragmentation.

(5) Part c) This allocation scheme is very fast for random access. Once we have the **FT**, any byte can be read with one additional block read (average=1, maximum=1).
The advantage of this allocation method is it allows for large file sizes and fast access with a small index table. The disadvantage is the external fragmentation.

(5) Part d) A linked allocation scheme is good for sequential access but not for random access. On **average** it will take (n+1)/2 **block reads** to read a single byte at a random position in the file. The expected value of an integer randomly selected from 1 to n is (n+1)/2. The **maximum** number of **block reads** that it takes to read a single byte in the file is n (in the last block).

(20) Part e)
```
int eDisk_ReadBlock(unsigned char *pt,    // result returned by reference
  unsigned short blockNum);               // which block to read
unsigned short FTbuf[64][2];   // place to store FT
unsigned char Databuf[256];    // place to store data
unsigned char eFile_Read(unsigned short numFT, unsigned long address){
unsigned long logical,byte,physical,i,j;
  eDisk_ReadBlock(FTbuf, numFT);  // read FT
  logical = address/256;          // logical block number
  byte = address&0xFF;            // byte in that block
  i = 0;             // search FT
  j = 0;
  while(1){
    if(i==64) return 0xff;          // error, byte does not exist
    if(FTbuf[i][0]==0) return 0xff; // error, byte does not exist
    for(k=FTbuf[i][0]; k<=FTbuf[i][1]; k++){
      if(j == logical){
        physical = k; break;
      }
      j++;
    }
    i++;
  }
  eDisk_ReadBlock(Databuf, physical);  // read data
  return Databuf[byte];
}
```