Agnihotri Engineering & GATE Classes

Scripting success stories

The following notes are from AEGC website http://www.aegc.yolasite.com

Questions Based Revision Sheet on **FIR AND IIR FILTERS**

Finite Impulse Response Filter FAQ

Q.1) What are "FIR filters"?

FIR filters are one of two primary types of digital filters used in Digital Signal Processing (DSP) applications (the other type being IIR).

Q.2) What does "FIR" mean?

"FIR" means "Finite Impulse Response".

Q.3) Why is the impulse response "finite"?

The impulse response is "finite" because there is no feedback in the filter; if you put in an impulse (that is, a single "1" sample followed by many "0" samples), zeroes will eventually come out after the "1" sample has made its way in the delay line past all the coefficients.

Q.4) What is the alternative to FIR filters?

DSP filters can also be "Infinite Impulse Response" (IIR). (IIR filters use feedback, so when you input an impulse the output theoretically rings indefinitely.

Q.5) How do FIR filters compare to IIR filters?

Each has advantages and disadvantages. Overall, though, the advantages of FIR filters outweigh the disadvantages, so they are used much more than IIRs.

Q.6) What are the *advantages* of FIR Filters (compared to IIR filters)?

Compared to IIR filters, FIR filters offer the following advantages:

- They can easily be designed to be "linear phase" (and usually are). Put simply, linear-phase filters delay the input signal, but don’t distort its phase.
- They are simple to implement. On most DSP microprocessors, the FIR calculation can be done by looping a single instruction.
- They are suited to multi-rate applications. By multi-rate, we mean either "decimation" (reducing the sampling rate), "interpolation" (increasing the sampling rate), or both. Whether decimating or interpolating, the use of FIR filters allows some of the calculations to be omitted, thus providing an important computational efficiency. In contrast, if IIR filters are used, each output must be individually calculated, even if it that output will discarded (so the feedback will be incorporated into the filter).
- They have desireable numeric properties. In practice, all DSP filters must be implemented using "finite-precision" arithmetic, that is, a limited number of bits. The use of finite-precision arithmetic in IIR filters can cause significant problems due to the use of feedback, but FIR filters have no feedback, so they can usually be implemented using fewer bits, and the designer has fewer practical problems to solve related to non-ideal arithmetic.

- They can be implemented using fractional arithmetic. Unlike IIR filters, it is always possible to implement a FIR filter using coefficients with magnitude of less than 1.0. (The overall gain of the FIR filter can be adjusted at its output, if desired.) This is an important considertaion when using fixed-point DSP's, because it makes the implementation much simpler.

Q.7) What are the *disadvantages* of FIR Filters (compared to IIR filters)?

Compared to IIR filters, FIR filters sometimes have the disadvantage that they require more memory and/or calculation to achieve a given filter response characteristic. Also, certain responses are not practical to implement with FIR filters.

Q.9) What terms are used in describing FIR filters?

- *Impulse Response* - The "impulse response" of a FIR filter is actually just the set of FIR coefficients. (If you put an "impluse" into a FIR filter which consists of a "1" sample followed by many "0" samples, the output of the filter will be the set of coefficients, as the 1 sample moves past each coefficient in turn to form the output.)
- *Tap* - A FIR "tap" is simply a coefficient/delay pair. The number of FIR taps, (often designated as "N") is an indication of 1) the amount of memory required to implement the filter, 2) the number of calculations required, and 3) the amount of "filtering" the filter can do; in effect, more taps means more stopband attenuation, less ripple, narrower filters, etc.)
- *Multiply-Accumulate (MAC)* - In a FIR context, a "MAC" is the operation of multiplying a coefficient by the corresponding delayed data sample and accumulating the result. FIRs usually require one MAC per tap. Most DSP microprocessors implement the MAC operation in a single instruction cycle.
- *Transition Band* - The band of frequencies between passband and stopband edges. The narrower the transition band, the more taps are required to implement the filter. (A "small" transition band results in a "sharp" filter.)
- *Delay Line* - The set of memory elements that implement the "Z^-1" delay elements of the FIR calculation.
- *Circular Buffer* - A special buffer which is "circular" because *incrementing* at the end causes it to wrap around to the beginning, or because *decrementing* from the beginning causes it to wrap around to the end. Circular buffers are often provided by DSP microprocessors to implement the "movement" of the samples through the FIR delay-line without having to literally move the data in memory. When a new sample is added to the buffer, it automatically replaces the oldest one.

Linear Phase

Q.10) What is the association between FIR filters and "linear-phase"?

Most FIRs are linear-phase filters; when a linear-phase filter is desired, a FIR is usually used.

Q.11) What is a *linear phase* filter?

"Linear Phase" refers to the condition where the phase response of the filter is a linear (straight-line) function of frequency (excluding phase wraps at +/- 180 degrees). This results in the *delay* through the filter being the same at all frequencies. Therefore, the filter does not cause "phase distortion" or "delay distortion". The lack of phase/delay distortion can be a critical advantage of FIR filters over IIR and analog filters in certain systems, for example, in digital data modems.

Q.12) What is the condition for linear phase?

FIR filters are usually designed to be linear-phase (but they don't have to be.) A FIR filter is linear-phase if (and only if) its coefficients are symmetrical around the center coefficient, that is, the first coefficient is the same as the last; the second is the same as the next-to-last, etc. (A linear-phase FIR filter having an odd number of coefficients will have a single coefficient in the center which has no mate.)

Q.13) What is the delay of a linear-phase FIR?

The formula is simple: given a FIR filter which has N taps, the delay is: (N - 1) / (2 * Fs), where Fs is the sampling frequency. So, for example, a 21 tap linear-phase FIR filter operating at a 1 kHz rate has delay: (21 - 1) / (2 * 1 kHz)=10 milliseconds.

Q.14) What is the alternative to linear phase?

Non-linear phase, of course. ;-) Actually, the most popular alternative is "minimum phase". Minimum-phase filters (which might better be called "minimum delay" filters) have less delay than linear-phase filters with the same amplitude response, at the cost of a non-linear phase characteristic, a.k.a. "phase distortion".

A lowpass FIR filter has its largest-magnitude coefficients in the center of the impulse response. In comparison, the largest-magnitude coefficients of a minimum-phase filter are nearer to the beginning.

Q.15) What is the Z transform of a FIR filter?

For an N-tap FIR filter with coefficients h(k), whose output is described by:

y(n)=h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + ... h(N-1)x(n-N-1),

the filter's Z transform is:

H(z)=h(0)z-0 + h(1)z-1 + h(2)z-2 + ... h(N-1)z-(N-1) , or

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$$

Q.16) What is the frequency response formula for a FIR filter?

The variable z in H(z) is a continuous complex variable, and we can describe it as: $z=r\cdot e^{j\omega}$, where r is a magnitude and ω is the angle of z. If we let r=1, then H(z) around the unit circle becomes the filter's frequency response H(jω). This means that substituting $e^{j\omega}$ for z in H(z) gives us an expression for the filter's frequency response H(ω), which is:

$H(j\omega)=h(0)e^{-j0\omega} + h(1)e^{-j1\omega} + h(2)e^{-j2\omega} + \dots h(N-1)e^{-j(N-1)\omega}$, or

Using Euler's identity, $e^{-ja}=\cos(a) - j\sin(a)$, we can write H(ω) in rectangular form as:

H(jω)=h(0)[cos(0ω) - jsin(0ω)] + h(1)[cos(1ω) - jsin(1ω)] + ... h(N-1)[cos((N-1)ω) - jsin((N-1)ω)] , or

$$H(j\omega) = \sum_{n=0}^{N-1} h(n)[\cos(n\omega) - j\sin(n\omega)]$$

Q.17) Can I calculate the frequency response of a FIR using the Discrete Fourier Transform (DFT)?

Yes. For an N-tap FIR, you can get N evenly-spaced points of the frequency response by doing a DFT on the filter coefficients. However, to get the frequency response of the filter at any *arbitrary* frequency (that is, at frequencies *between* the DFT outputs), you will need to use the formula above.

Q.18) What is the DC gain of a FIR filter?

Consider a DC (zero Hz) input signal consisting of samples which each have value 1.0. After the FIR's delay line had filled with the 1.0 samples, the output would be the sum of the coefficients. Therefore, the gain of a FIR filter at DC is simply the sum of the coefficients.

This intuitive result can be checked against the formula above. If we set ω to zero, the cosine term is always 1, and the sine term is always zero, so the frequency response becomes:

$$H(j\omega) = \sum_{n=0}^{N-1} h(n)$$

Q.19) How do I scale the gain of a FIR filter?

Simply multiply all coefficients by the scale factor.

Numeric Properties

Q.20) Are FIR filters inherently stable?

Yes. Since they have no feedback elements, any bounded input results in a bounded output.

Q.21) What makes the numerical properties of FIR filters "good"?

Again, the key is the lack of feedback. The numeric errors that occur when implementing FIR filters in computer arithmetic occur separately with each calculation; the FIR doesn't "remember" its past numeric errors. In contrast, the feedback aspect of IIR filters can cause numeric errors to compound with each calculation, as numeric errors are fed back.

The practical impact of this is that FIRs can generally be implemented using fewer bits of precision than IIRs. For example, FIRs can usually be implemented with 16 bits, but IIRs generally require 32 bits, or even more.

Q.22) Why are FIR filters generally preferred over IIR filters in multirate (decimating and interpolating) systems?

Because only a fraction of the calculations that would be required to implement a decimating or interpolating FIR in a literal way actually needs to be done.

Since FIR filters do not use feedback, only those outputs which are actually going to be used have to be calculated. Therefore, in the case of decimating FIRs (in which only 1 of N outputs will be used), the other N-1 outputs don't have to be calculated. Similarly, for interpolating filters (in which zeroes are inserted between the input samples to raise the sampling rate) you don't actually have to multiply the inserted zeroes with their corresponding FIR coefficients and sum the result; you just omit the multiplication-additions that are associated with the zeroes (because they don't change the result anyway.)

In contrast, since IIR filters use feedback, every input must be used, and every input must be calculated because all inputs and outputs contribute to the feedback in the filter.

Q.23) What special types of FIR filters are there?

Aside from "regular" and "extra crispy" there are:

- *Boxcar* - Boxcar FIR filters are simply filters in which each coefficient is 1.0. Therefore, for an N-tap boxcar, the output is just the sum of the past N samples. Because boxcar FIRs can be implemented using only adders, they are of interest primarily in hardware implementations, where multipliers are expensive to implement.
- *Hilbert Transformer* - Hilbert Transformers shift the phase of a signal by 90 degrees. They are used primarily for creating the imaginary part of a complex signal, given its real part.
- *Differentiator* - Differentiators have an amplitude response which is a linear function of frequency. They are not very popular nowadays, but are sometimes used for FM demodulators.
- *Lth-Band* - Also called "Nyquist" filters, these filters are a special class of filters used primarily in multirate applications. Their key selling point is that one of every *L* coefficients is zero--a fact which can be exploited to reduce the number of multiply-accumulate operations required to implement the filter. (The famous "half-band" filter is actually an Lth-band filter, with L=2.)
- *Raised-Cosine* - This is a special kind of filter that is sometimes used for digital data applications. (The frequency response in the passband is a cosine shape which has been "raised" by a constant.)

Infinite Impulse Response Filter FAQ

Q.24) What are IIR filters? What does "IIR" mean?

IIR filters are one of two primary types of digital filters used in Digital Signal Processing (DSP) applications (the other type being FIR). "IIR" means "Infinite Impulse Response".

Q.25) Why is the impulse response "infinite"?

The impulse response is "infinite" because there is feedback in the filter; if you put in an impulse (a single "1" sample followed by many "0" samples), an infinite number of non-zero values will come out (theoretically).

Q.26) What is the alternative to IIR filters?

DSP filters can also be "Finite Impulse Response" (FIR). FIR filters do not use feedback, so for a FIR filter with *N* coefficients, the output always becomes zero after putting in *N* samples of an impulse response.

Q.27) What are the advantages of IIR filters (compared to FIR filters)?

IIR filters can achieve a given filtering characteristic using less memory and calculations than a similar FIR filter.

Q.28) What are the disadvantages of IIR filters (compared to FIR filters)?

- They are more susceptible to problems of finite-length arithmetic, such as noise generated by calculations, and limit cycles. (This is a direct consequence of feedback: when the output isn't computed perfectly and is fed back, the imperfection can compound.)
- They are harder (slower) to implement using fixed-point arithmetic.
- They don't offer the computational advantages of FIR filters for multirate (decimation and interpolation) applications