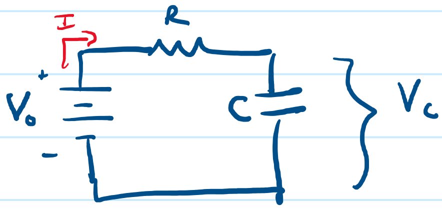
Discrete-Time LTI Systems

Sunday, August 14, 2022 2:07 PM

Difference Equations

Analog system behavior is specified by differential equations. Analog LTI systems are specified by Linear differential equations.



Q: charge on capacitor

$$V_0 - IR - \frac{Q}{C} = 0$$

$$\boxed{V_0 - \frac{dQ}{dt}R - \frac{Q}{C} = 0}$$

$$V_0 - \frac{Q}{C} = \frac{dQ}{dt}R$$

$$\frac{V_0}{R} - \frac{Q}{RC} = \frac{dQ}{dt}$$

$$\frac{CV_0 - Q}{RC} = \frac{dQ}{dt}$$

$$\frac{dQ}{CV_0 - Q} = \frac{dt}{RC}$$

$$\int^{q(t)} \frac{dQ}{CV_0 - Q} = \int^{t} \frac{dt}{RC}$$

$$\int_0 \frac{dQ}{CV_0 - Q} = \int_0 \frac{dt}{RC}$$

$$\ln(CV_0 - Q)\Big]_0^{q(t)} = -\frac{t}{RC} + \underbrace{C'}_{\text{integration constant}}$$

$$\ln(CV_0 - q(t)) - \ln(CV_0) =$$

$$\ln\left(\frac{CV_0 - q(t)}{CV_0}\right) = -t/RC + C'$$

$$1 - \frac{q(t)}{CV_0} = e^{C'} e^{-t/RC}$$

$$\frac{q(t)}{CV_0} = 1 - e^{C'} e^{-t/RC}$$

$$q(t=0) = 0 \Rightarrow e^{C'} = 1.$$

$$q(t) = CV_0(1 - e^{-t/RC})$$

$$I(t) = \frac{dq(t)}{dt} = \frac{CV_0}{RC} e^{-t/RC} = \frac{V_0}{R} e^{-t/RC}$$

We can see that, for example, a linear ODE completely specifies the behavior of an analog RC circuit. In discrete-time circuits, we use difference equations to specify LTI DT systems.

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + \cdots + a_p y[n-p]$$
$$+ b_0 x[n] + b_1 x[n-1] + \cdots + b_q x[n-q]$$

We specify the initial conditions of an LTI DT system as 0.
Why? To guarantee linearity.

Proof:

If the initial conditions of the output are not chosen as 0, then a system w/ non-zero difference equation coefficients may map a 0-input to an non-zero output, violating the linearity condition that

$$S\{\alpha x[n]\} = \alpha S\{x[n]\} \text{ for } \alpha = 0.$$

Example $y[n] = a y[n-1] + b x[n]$

n	$x[n]$	$y[n]$
-1	0	0
0	1	b
1	0	ab
2	0	$a^2 b$
3	0	$a^3 b$
4	0	$a^4 b$
⋮	⋮	⋮

Find $S\{\delta[n]\}$.

Systems w/ an infinitely non-zero response to an impulse input (like the example) are called IIR filters (Infinite Impulse Response).

Systems w/ a finite non-zero response to an impulse input ($a_1, \ldots, a_p = 0$) are called FIR filters (finite impulse response).

impulse input ($a_1, \ldots, a_p = 0$) are called FIR filters (finite impulse response).

DT Filters in Frequency Domain

$$x[n] = X e^{j2\pi fn}$$

$$y[n] = Y e^{j2\pi fn}$$

$$Y e^{j2\pi fn} = \sum_{m=1}^{p} a_m Y e^{j2\pi f(n-m)} + \sum_{m=0}^{q} b_m X e^{j2\pi f(n-m)}$$

$$Y \left(1 - \sum_{m=1}^{q} a_m e^{-j2\pi fm}\right) = \left(\sum_{m=0}^{q} b_m e^{-j2\pi fm}\right) X$$

$$\boxed{H(e^{j2\pi f}) = \frac{Y}{X} = \frac{\sum_{m=0}^{q} b_m e^{-j2\pi fm}}{1 - \sum_{m=1}^{q} a_m e^{-j2\pi fm}}}$$

This relation can also be derived from Fourier Transforms.

$y[n] = S\{x[n]\}$ ← (Take DTFT of both sides)

The unit sample response $h[n]$ is the output of LTI system S for input $\delta[n]$.

$$\mathcal{F}\{\delta[n]\} = \sum_{n=-\infty}^{n=+\infty} \delta[n] e^{-j2\pi fn} = 1$$

$$h[n] = \mathcal{F}^{-1}\left\{H(e^{j2\pi f}) \cdot \mathcal{F}\{\delta[n]\}\right\}$$

$$= \mathcal{F}^{-1}\{H(e^{j2\pi f})\}.$$

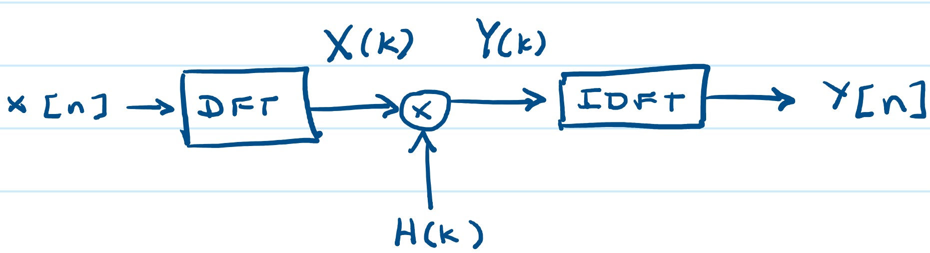
We have shown $h[n] \xleftrightarrow{\mathcal{F}} H(e^{j2\pi f})$

We can use the DFT to perform filtering in the discrete-time frequency domain.

$$H(k) \equiv H(e^{j\frac{2\pi f k}{N}}) \text{ for } k \in [0, 1, \ldots, N-1]$$

The IDFT will always be periodic with period N, so if $h[n]$ is longer than N, filtering in the frequency will not work as aliasing in the time domain will occur.

So for FIR filters, we require $N \geq Q$.
IIR filters cannot be implemented in the frequency domain.

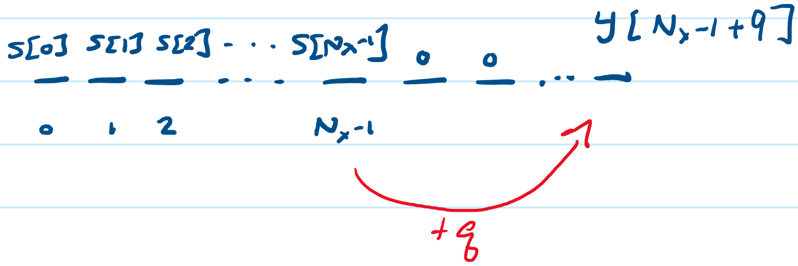


Above shows how to implement an FIR filter in the frequency domain. There is an additional sublety, however.

Suppose we input a signal of length N_x into an FIR filter with length $q+1$.

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_q x[n-q]$$

$$s[n] \longrightarrow \boxed{S} \longrightarrow S\{s[n]\}$$



The last non-zero (potentially) output occurs when $n = N_x - 1 + q$. $S\{s[n]\}$ has length $N_x + q$.

To prevent time-domain aliasing, we require $N_x + q$ samples of $H(k)$ and $X(k)$.

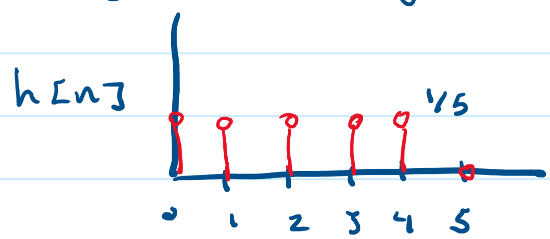
$$X(k) = \sum_{n=0}^{N_x-1} s[n] e^{-j\frac{2\pi k n}{N_x}}$$

$$X(k) = X(k+N_x).$$

To sample at $N_x + q$ frequencies, we need an N_x+q-length input signal w/ the same behavior as the original one. So we zero-pad both $h[n]$ and $x[n]$.

DJIA Averaging

Let's look at an example use-case. Suppose we want to compute a running average of DJIA values. For a year, we have 253 stock prices. We will use a length-5 averager.



The output will be length 258, so we must use a 512-length FFT.

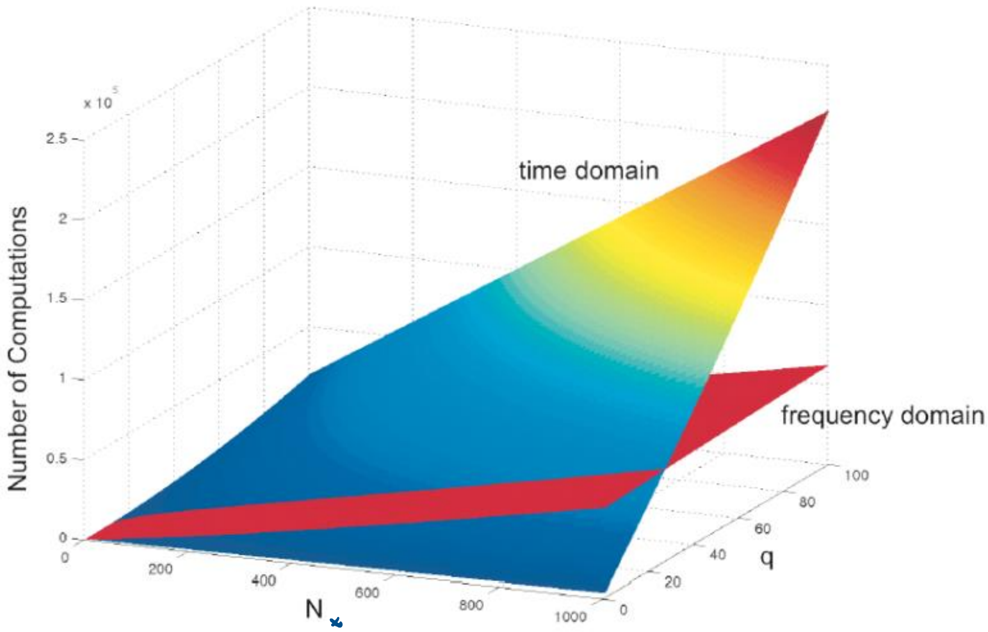
$\boxed{\text{Complexity in Time and Frequency Domains}}$

For a q-order FIR filter, time-domain filtering requires $(N_x+q)(2q+1)$ operations. We must compute N_x+q potentially non-zero outputs, and each requires q multiplications and $q+1$ additions or $2q+1$ total arithmetic operations.

For a q-order FIR filter, frequency-domain filtering requires $\underbrace{\left(\frac{5K}{2}\log_2(K)\right)\cdot 2}_{\text{FFTs}} + \underbrace{6K}_{H(k)X(k)}$ where $K = N_x + q$.

$5(N_x+q)\log_2(N_x+q) + 6(N_x+q)$ is the total number of frequency-domain operations required.

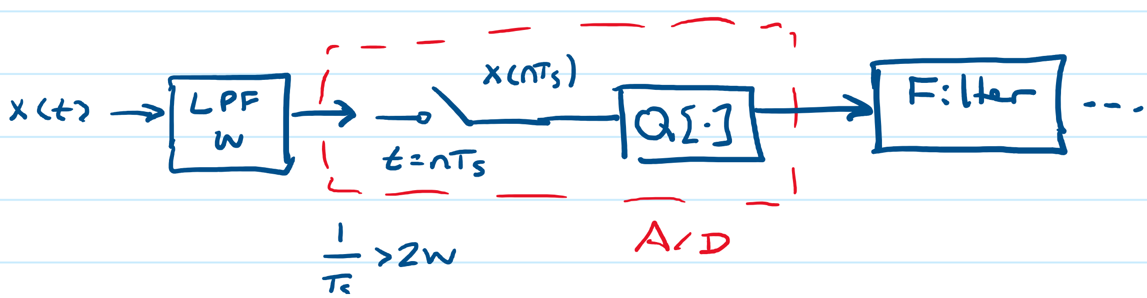
Plotting # of computations vs. q and N_x yields the following figure from the textbook:



Filtering Infinite-Length Inputs

To filter an infinite-length signal, we section it into N_x-length non-overlapping sections and filter each section, combining the results at the end thanks to linearity.

Computationally Filtering Analog Signals



$$\frac{1}{T_s} > 2W$$

A/D