Difference equation

$$y(n) = \sum_{i=0}^{N} b_i x(n-i) + \sum_{j=1}^{M} a_j y(n-M)$$

If $a_j >_0$ for some j, then the filter is called recursive, or an infinite impulse **IIR** filter
Else, there is no feedback and the filter is a finite impulse response **FIR** filter. This equation only represents causal filters.

Impulse response

$$h(n) \triangleq \mathcal{L}_n\{\delta(.)\}$$

Where $\delta = [1, 0, 0, 0, \ldots]$
Any LTI filter can be applied by convolving the signal with the filter's impulse response $h(n)$.
The impulse response has to decay to zero as $n \to \infty$, otherwise it is unstable

Convolution

$$f * g(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau)\, d\tau$$

Think of this as the entirety of f being weighted by the reflection of g. This reflection is shifted to the right (towards infinity) one time-step at a time.

Representing Filter via Impulse

Each sample in signal $x(n)$ can be thought of as a scaled and shifted impulse response. The entirety of the signal can be represented as the sum of scaled and shifted impulse responses. Remember that LTI filters have scalability, superposition, and time-invariance properties. Therefore, we can scale and shift the impulse response of the filter according to the value and index of each $x(n)$, and superimpose the results to get $y(n)$, which is equivalent to applying the filter to the signal.

Mathematical definition

A signal can be represented via its' convolution with an impulse response.

$$x(n) = \sum_{i=0}^{n} x(i)\delta(i-n) \triangleq (x * \delta)(n)$$

This might seem redundant, but it shows that we can express $x(n)$ as a linear scaled sum of impulses. Furthermore, using the definition $h(n) \triangleq \mathcal{L}_n\{\delta(.)\}$, we can derive:

$$
\begin{aligned}
\mathcal{L}_n\{x(.)\} &= \mathcal{L}_n\{(x*\delta)(.)\} \\
&= \mathcal{L}_n\left\{\sum_{i=-\infty}^{\infty} x(i)\delta(.-i)\right\} \\
&= \sum_{i=-\infty}^{\infty} x(i)\mathcal{L}_n\{\delta(.-i)\} \\
&\triangleq \sum_{i=-\infty}^{\infty} x(i)h(n,i)
\end{aligned}
$$

Here, think of $h(n,i)$ as the filter response at time n to an impulse that occurred at time i. So at each time-step n, the output value is equal to the scaled sum of shifted impulses.

If in addition to being linear, the filter is also time-invariant, then we can write $h(n,i) = h(n-i)$, which gives us:

$$y(n) = \sum_{i=-\infty}^{\infty} x(i)h(n-i) \triangleq (x*h)(n), \quad n \in Z$$

Assuming the filter is linear, time-invariant *and* causal:

$$y(n) = \sum_{i=0}^{n} x(i)h(n-i) \triangleq (x*h)(n), \quad n \in [0,1,2,\ldots]$$

Since convolution is commutative, we also have:

$$y(n) = \sum_{i=0}^{n} h(i)x(n-i) = (x*h)(n), \quad n \in [0,1,2,\ldots]$$

or

$$y(n) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2)\ldots h(n)x(0)$$

It is also evident that the filter operates by summing *weighted echoes* of the input signal together. At time n, the weight of the echo from i samples ago is $x(n-i)$ is $h(i)$

Signal flow graphs

See chapter 9.
https://ccrma.stanford.edu/~jos/filters/Signal_Flow_Graph_I.html
https://ccrma.stanford.edu/~jos/filters/Direct_Form_II.html

Further reading

Physical modeling book: https://ccrma.stanford.edu/~jos/pasp/Finite_Difference_Schemes.html

Questions

When convolving two functions/signals which do not have a non-zero value until time t, does it take 2t timesteps before we see a non-negative value?