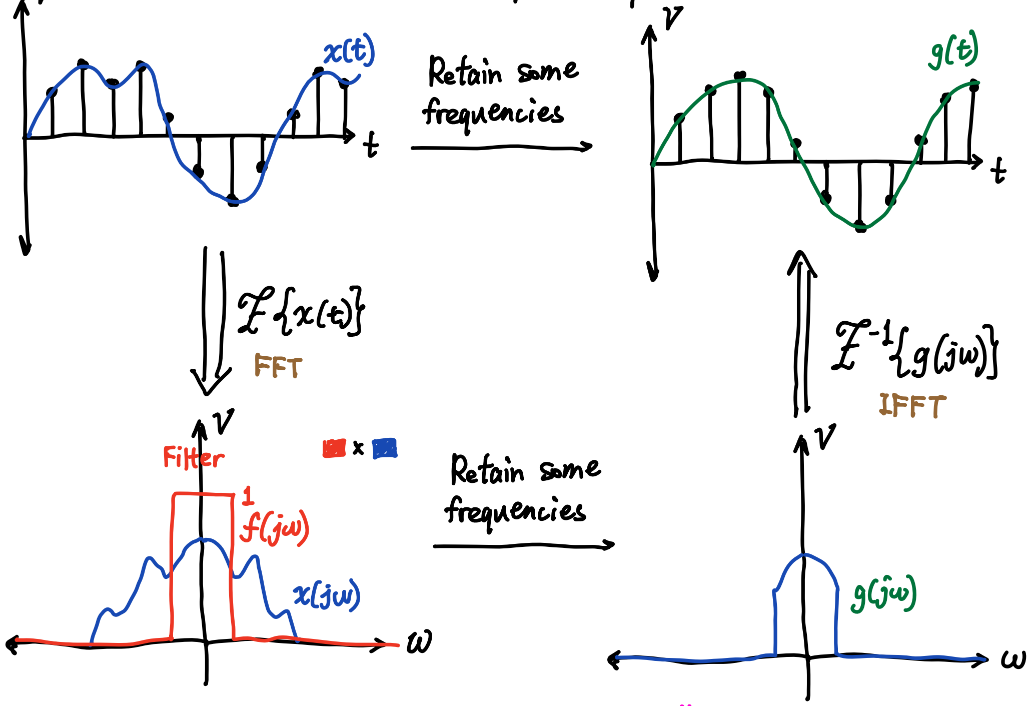
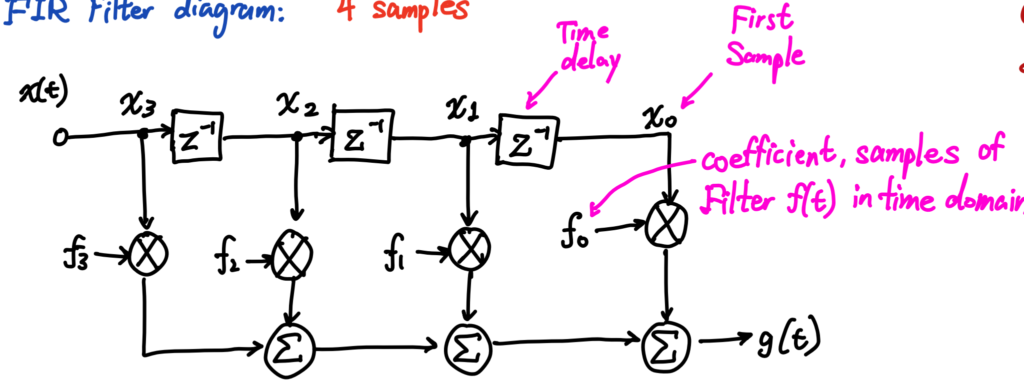
Finite Impulse Response Filter



$x(j\omega)\cdot f(j\omega) = \boxed{x(t) * f(t)}$ FIR filter does this.

FIR Filter diagram: 4 samples



Convolution: Multiplication of one signal by another and the integrating that product.

Because number of f_i is finite, filter can't be ideal. (Trade passband ripple vs. roll off slope)

$y[n] = \sum_{k=0}^{N} b_k\, x[n-k]$ $\quad H(z) = \sum_{k=0}^{N} b_k z^{-k} = b_0 \cdot \prod_{k=1}^{N} (1 - c_k z^{-1})$ $\quad$ Only pole at $z=0$.

$h[n] = \begin{cases} b_n, & 0 \le n \le N \\ 0, & \text{otherwise} \end{cases}$ $\quad$ Optimize $\underset{\text{This class}}{\underline{\text{RMSE}}}$ or Maximum Error : $H(e^{j\omega}) \Leftrightarrow \overset{\text{desired}}{H_d(e^{j\omega})}$

FIR can have arbitrary magnitude/phase response, can obtain linear phase.

FIR pseudocode:

```
Load coefficient
while core_enable is True:
    If reset is active:
        reset x
    End if
    for i=63, i>0, i=i-1:
        x[i]=x[i-1]
    end for
    If input_end is not active:
        x[0] = d_in
    End if
    Else:
        x[0] = 0
    for i=63, i≥0, i=i-1:
        sum += b[63-i]·x[i]
    end for
    output sum
    If x is empty AND input_end is active:
        core_enable = False
    End if
End while
```