Signals

A discrete time signal can be described as a mathematical function $x[n]$ where n is the index of the sample. Or as an Array/List of the significant samples, with any sample not listed being 0. Or by plotting. We can modify a signal by multiplying it by something or adding something to it. We can also modify the signal by time shifting it: ie $X(t-3)$ will delay the signal by shifting it to the right, and $X(t+3)$ will advance the signal by shifting it to the left. Likewise we can multiply the signal by a constant $c > 1$ will effectively "downsample" the signal, and multiplying it by $c < 1$ will "upsample" the signal. Likewise we can time reverse the signal.

Delta and Unit Step Signals

We define the **delta signal** as $\boxed{\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}}$. From this we have the following sampling property: $\boxed{x[n] \cdot \delta[n-k] = x[k]\delta[n-k]}$.

From this delta signal we can define the **unit step** signal $\boxed{u[n] = \sum_{k=0}^{\infty} \delta[n-k] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}}$.

Periodicity

A signal is periodic if it can be written as $x[n] = x[n+N]$ for some integer N and all n. A signal's fundamental period is the smallest integer N such that $x[n] = x[n+N]$ for all n.

Even and Odd Signals

A signal is even if $x[n] = x[-n]$. A signal is odd if $x[n] = -x[-n]$. We can decompose any singal into its even and odd parts with the even part $x_e[n] = \frac{1}{2}(x[n] + x[-n])$ and the odd part $x_o[n] = \frac{1}{2}(x[n] - x[-n])$.

Energy and Power Signals

We define the energy of a signal as $\boxed{E = \sum_{n=-\infty}^{\infty} |x[n]|^2}$. We define the power of a signal as $\boxed{P = \lim_{m\to\infty} \frac{1}{2m+1} \sum_{n=-\infty}^{M} |x[n]|^2}$. If the signal is periodic, this can be simplified to $\boxed{P = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2}$. We call a signal a **energy signal** if its energy is finite, and a **power signal** if its power is finite. A power signal will have infinite energy.

System Properties

A system is **linear** if given the outputs $y_1[n]$, and $y_2[n]$ of two inputs $x_1[n]$ and $x_2[n]$ respectively, then the output of the input $x_3 = \alpha x_1[n] + \beta x_2[n]$ the output is $y_3[n] = \alpha y_1[n] + \beta y_2[n]$. A system is **time invariant** if given the output $y[n]$ of an input $x[n]$, then the output of the input $x[n-k]$ is $y[n-k]$. A system is **causal** if the output of an input $x[n]$ is only dependent on the input $x[n]$ and not $x[n-k]$ for $k < 0$. A system is **stable** if the output of an input $x[n] < \alpha < \infty$ for all n, the output $y[n] < \beta < \infty$ for all n. A system is **relaxed** if $y[n] \to 0$ for $n \to \infty$ and 0 when the input is 0. In general for a system that is lti, given its impulse response $h[n]$ (the system response to $\delta[n]$), the output of an input $x[n]$ is given by $y[n] = x[n] * h[n]$ and the system will be stable if and only if $\sum_{k=-\infty}^{\infty} |h[k]| < \infty$.

Convolution

We define the convolution of two signals $x[n]$ and $h[n]$ as $\boxed{y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]}$. A shortcut on how to do convolution is shown below:

We have that if given two signals $x_1[n]$ and $x_2[n]$ that start at n_1 and n_2 respectively and have lengths of L_1 and L_2 respectively, then the convolution of the two signals will have nonzeros values for $n_1 + n_2 \leq n \leq n_1 + n_2 + L_1 + L_2$, and thus the convolution will have $L_1 + L_2 + 1$ nonzero values. $u[n] * u[n] = r[n+1]$ where $r[n] = nu[n]$ is the unit ramp signal.

➲ The "n" dependency of $y[n]$ deserves some care: for each value of "n" the convolution sum must be computed ***separately*** over all values of a dummy variable "m". So, for each "n"

1. Rename the independent variable as m. You now have $x[m]$ and $h[m]$. Flip $h[m]$ over the origin. This is $h[-m]$
2. Shift $h[-m]$ as far left as possible to a point "n", where the two signals barely touch. This is $h[n-m]$
3. Multiply the two signals and sum over all values of m. This is the convolution sum for the specific "n" picked above.
4. Shift / move $h[-m]$ to the right by one sample, and obtain a new $h[n-m]$. Multiply and sum over all m.
5. Repeat 2~4 until $h[n-m]$ no longer overlaps with $x[m]$, i.e., shifted out of the $x[m]$ zone.

Discrete Time Fourier Transform

If a signal is periodic with period N, then we have that we can represnt it as a discrete fourier series, $\boxed{x[n] = \frac{1}{N} \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi}{N}kn}}$.

Where c_k is derived from $\boxed{c_k = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}}$.

Aditional Notes

conjugate symmetric: $x^*[n] = x[-n]$ **conjugate anti-symmetric:** $x^*[n] = -x[-n]$. $\boxed{\sum_{k=1}^{n} ar^{k-1} = \begin{cases} \frac{a(1-r^n)}{1-r} & r \neq 1 \\ an & r = 1 \end{cases}}$.

The sum of an arithmetic sequence consisting of n values is $\boxed{\frac{n(a_1 + a_n)}{2}}$

Periodic Convolution

Given that two signals $\tilde{x}[n]$ and $\tilde{y}[n]$ are periodic with period N, the we have that the periodic convolution of the two signals is given by $\boxed{y[n] = \tilde{x}[n] \bigotimes \tilde{y}[n] = \sum_{k=0}^{N-1} \tilde{x}[k]\tilde{y}[n-k]}$ it is important to not that $\tilde{x}[n] \bigotimes \tilde{y}[n] = \tilde{x}[n] \bigotimes \tilde{y}[n-N] \neq \tilde{x}[n] * \tilde{y}[n]$.

Discrete Time Fourier Transform

We have that the DFTF in general of a signal is: $X_{2\pi}(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$ And its corresponding inverse DTFT is $x[n] = \frac{1}{2\pi}\int_{2\pi} X_{2\pi}(\omega)e^{j\omega n}d\omega$

Discrete Fourier Tranform

Given a signal with a period of N, the discrete fourier transform of the signal is $X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}$ for $0 \leq k \leq N-1$. And its inverse is, and with an inverse fourier transform of $x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi}{N}kn}$. We can view the DFT as a sampling of the DTFT with $X[k] = X_{2\pi}(\frac{2\pi}{N}k)$,

Linear Algebra for the DFT

We can express the DFT as a matrix multiplication, where the vector X_k is the DFT of the signal represented by the vector x_n of size n then we have that letting $W = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \omega^2 & \cdots & \omega^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix}$ we get that $\boxed{X_k = Wx_n}$, where $\omega = e^{-j\frac{2\pi}{N}}$. An important thing to note about W is that it is symetric, therefore the inverse fourier transform is just $\boxed{x_n = \frac{1}{N}W^H X_k}$ where W^H is the conjugate transpose of W, or since W is symetric, $W^H = W^*$, where W^* is the conjugate of W.

FFT

The naive implementation of the DFT is $O(N^2)$, however we can use the property that we can rewrite the DFT as $X_k = \sum_{m=0}^{N/2-1} x_{2m}e^{-j\frac{2\pi}{N}k}$ $\sum_{m=0}^{N/2-1} x_{2m+1}e^{-j\frac{2\pi}{N}(2m+1)k} = \sum_{m=0}^{N/2-1} x_{2m}e^{-j\frac{2\pi}{N}k2m} + e^{-j\frac{2\pi}{N}k}\sum_{m=0}^{N/2-1} x_{2m+1}e^{-j\frac{2\pi}{N}(2m)k}$ Therefore we can break the DFT into two smaller DFTs, and then combine them to get the DFT of the original signal. repeating this we get that we can optimize the FFT's complexity to $O(NlogN)$, where N is the length of the signal.

Z transform

The z transform of a signal is given by $X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$. We define the reigon of convergence as the set of z such that this sum exists.