Systems in the Time-Domain

Bởi:

Don Johnson

A discrete-time signal $s(n)$ is delayed by n_0 samples when we write $s(n-n_0)$, with $n_0>0$. Choosing n_0 to be negative advances the signal along the integers. As opposed to analog delays, discrete-time delays can **only** be integer valued. In the frequency domain, delaying a signal corresponds to a linear phase shift of the signal's discrete-time Fourier transform: $\leftrightarrow\left(s(n-n_0),\ e^{-i2\pi f n_0}\left(S\left(e^{i2\pi f}\right)\right)\right)$.

Linear discrete-time systems have the superposition property.

Superposition
$S\left(a_1(x_1(n))+a_2(x_2(n))\right)=a_1\left(S(x_1(n))\right)+a_2\left(S(x_2(n))\right)$
A discrete-time system is called shift-invariant (analogous to time-invariant analog systems) if delaying the input delays the corresponding output.
Shift-Invariant
If $S(x(n))=y(n)$, Then $S\left(x(n-n_0)\right)=y(n-n_0)$
We use the term shift-invariant to emphasize that delays can only have integer values in discrete-time, while in analog signals, delays can be arbitrarily valued.

We want to concentrate on systems that are both linear and shift-invariant. It will be these that allow us the full power of frequency-domain analysis and implementations. Because we have no physical constraints in "constructing" such systems, we need only a mathematical specification. In analog systems, the differential equation specifies the input-output relationship in the time-domain. The corresponding discrete-time specification is the difference equation.

The Difference Equation
$y(n)=a_1(y(n-1))+\ldots+a_p(y(n-p))+b_0(x(n))+b_1(x(n-1))+\ldots+b_q(x(n-q))$
Here, the output signal $y(n)$ is related to its **past** values $y(n-l)$, $l=\{1,\ \ldots,\ p\}$, and to the current and past values of the input signal $x(n)$. The system's characteristics are determined by the choices for the number of coefficients p and q and the coefficients' values $\{a_1,\ \ldots,\ a_p\}$ and $\{b_0,\ b_1,\ \ldots,\ b_q\}$. There is an asymmetry in the coefficients: where is a_0 ? This coefficient would multiply the $y(n)$ term in the difference equation.

We have essentially divided the equation by it, which does not change the input-output relationship. We have thus created the convention that a_0 is always one.

As opposed to differential equations, which only provide an **implicit** description of a system (we must somehow solve the differential equation), difference equations provide an **explicit** way of computing the output for any input. We simply express the difference equation by a program that calculates each output from the previous output values, and the current and previous inputs.