Systems in the Time-Domain*

Don Johnson

Abstract

Discrete-time systems allow for mathematically specified processes like the difference equation.

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[1], 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: $s(n-n_0) \leftrightarrow e^{-(i2\pi f n_0)} S\left(e^{i2\pi f}\right)$.

Linear discrete-time systems have the superposition property.

Superposition

$$S(a_1 x_1(n) + a_2 x_2(n)) = a_1 S(x_1(n)) + a_2 S(x_2(n)) \quad (1)$$

A discrete-time system is called **shift-invariant** (analogous to time-invariant analog systems) if delaying the input delays the corresponding output.

Shift-Invariant

$$\text{If} \quad S(x(n)) = y(n)\,,\ \text{Then} \quad S(x(n-n_0)) = y(n-n_0) \quad (2)$$

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

$$y(n) = a_1 y(n-1) + \dots + a_p y(n-p) + b_0 x(n) + b_1 x(n-1) + \dots + b_q x(n-q) \quad (3)$$

Here, the output signal $y(n)$ is related to its **past** values $y(n-l)$, $l=\{1,\dots,p\}$, 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 $\{a_1,\dots,a_p\}$ and $\{b_0,b_1,\dots,b_q\}$.

> ASIDE: There is an asymmetry in the coefficients: where is a_0 ? This coefficient would multiply the $y(n)$ term in the difference equation (3: 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.

*Version 2.7: Sep 2, 2004 7:21 pm -0500

†http://creativecommons.org/licenses/by/1.0

[1]"Simple Systems": Section Delay <http://cnx.org/content/m0006/latest/#delay>

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.