Effects of sampling: details

Consider the following system:

\[
\begin{array}{c}
 r(t) \quad \mathrm{S\&H} \quad C(s) \quad G(s) \quad o(t)
\end{array}
\]

We haven’t done anything digital here yet, just put a sample and hold upstream of two analog subsystem blocks, \(G(s)\) and \(C(s)\). All signals at all points are analog. Just to the right of the S\&H, you can describe the stair-stepped sample\&hold approximation of \(r(t)\) as

\[
\bar{r}(t) = r(0)[u(t) - u(t - T)] + r(T)[u(t - T) - u(t - 2T)] + \ldots r(nT)[u(t - nT) - u(t - [n + 1]T)].
\]

where \(T\) is the sampling period and \(u\) is the unit step function.

This analytic, but messy, function of time has a LaPlace Transform,

\[
\bar{R}(s) = r(0)\left[\frac{1}{s} - \frac{1}{s} e^{-sT}\right] + r(T)\left[\frac{1}{s} e^{-sT} - \frac{1}{s} e^{-2sT}\right] + \ldots = \frac{1 - e^{-sT}}{s} \sum_{n=0}^{\infty} r(nT)e^{-nT}.
\]

The transform \(\bar{R}^*(s)\) (the sum in the above equation) is the LaPlace transform of a train of pulses, separated in time by \(T\), whose heights are the values of the sampled sequence, \(r(nT)\).

Now, compare this result to the z-transform of the same sequence,

\[
R^*(s) = \sum_{n=0}^{\infty} r(nT)e^{-nT} \quad \text{and} \quad R(z) = \sum_{n=0}^{\infty} r(nT)z^{-n}.
\]

The LaPlace transform and the z-transform correspond exactly, if \(z = e^{sT}\).

To study the entirely analog (but also sampled) system in the diagram, you must work in the z-domain because of the non-linear nature of the sample and hold operation (which you can see above by the presence of non-polynomial terms like \(\exp\{sT\}\)). The transfer function of this system in the z-domain is

\[
H(z) = \mathcal{Z}\left\{\frac{1 - e^{-sT}}{s} C(s)G(s)\right\} = (1 - z^{-1})\mathcal{Z}\left\{\frac{C(s)G(s)}{s}\right\} = \frac{z - 1}{z} \mathcal{Z}\left\{\frac{C(s)G(s)}{s}\right\}.
\]

In this equation, \(\mathcal{Z}\{F(s)\}\) means the function of \(z\) “equivalent to” \(F(s)\). You can see from these results that is isn’t enough just to find the z-transform that is equivalent to the
LaPlace transform of a given system transfer function. You must explicitly take account of the sample-and-hold. Its presence is real.

In both the s- and the z- domains, the transfer function of a system is the transform (Laplace or z-) of the system’s impulse response. For continuous systems, this response is a continuous function of time, while for discrete systems it is a sequence of samples of a continuous function which is the “underlying” impulse response. Thus, expanding \( F(s) \) into the residue-pole form shows that the impulse response of any continuous linear system is a sum of exponential functions of time: if

\[
H(s) = \sum_i \frac{A_i}{s - p_i}, \quad \text{then} \quad h(t) = \sum_i A_i e^{p_i t},
\]

where the sum runs over all the poles. In the z-domain, we have already shown that the z-transform of an exponential time sequence is given by the z-transform pair,

\[
A e^{-aT} \longleftrightarrow A \frac{z}{z - e^{-aT}}.
\]

Now, compare the LaPlace and the z-transforms for an exponential function or sequence:

\[
\frac{A}{s + a} \longleftrightarrow A e^{-at} \longleftrightarrow A \frac{z}{z - e^{-aT}}.
\]

Since the time responses of both continuous and sampled systems can be expressed as a sum of exponential time functions (continuous) or sequences (sampled), this comparison suggests the following procedure for finding \( \mathcal{Z}\{F(s)\} \):

1. Put \( F(s) \) into residue-pole form, using a partial fraction expansion, or some equivalent approach.
2. For each term in the residue-pole sum, write the equivalent term in the z-domain using the known values for the residue, \( A \), and the pole, \( -a \).
3. Sum the terms in the z-domain to get a rational polynomial in \( z \).

Doing this procedure by hand is very tedious and error-prone. Fortunately, a numerical algorithm is available.

Another interesting (and potentially dangerous) property of z-transforms is that they can’t be multiplied for cascaded systems.

Sampling speed is usually at a premium in sampled-data control systems that use computers in the loop. Having the computer sample both the command signal, \( r(t) \), and the feedback signal cuts the available sampling frequency in half. Consequently, most single-input-single-output (SISO) computer-based feedback control systems are of the unity-feedback type, where the feedback transfer function is unity and the sampling is performed after the command and the feedback signals are differenced. For such a system with a digital cascade controller, \( D(z) \), the transfer function is

\[
H(z) = \frac{D(z) \mathcal{Z}\{G(s)\}}{1 + D(z) \mathcal{Z}\{G(s)\}}.
\]
The system block diagram in this case is

For designers with some insights what sort of cascade compensator, $C(s)$, might work in the system below,

it is often convenient to start with the digital equivalent of $C(s)$, and then reduce the sampling frequency while making modifications in $C(s)$ to maintain system performance.

**Caveats:**
1) Even if the “equivalence” condition,

$$D(z) = \frac{\sum \left\{ \frac{C(s)G(s)}{s} \right\}}{\sum \left\{ \frac{G(s)}{s} \right\}}$$

is satisfied, the two unity feedback systems pictured above perform alike only at high sampling frequencies.
2) To find digital equivalents of cascaded continuous systems, multiply the s-domain transfer functions first, THEN transform to the z-domain.