Lecture 06.03 Discrete transfer functions

We begin with a review of Laplace transforms and continuous transfer functions.

06.03.1 Laplace transforms

Laplace transform

In the analysis of this continuous systems, we use the *Laplace transform*, defined by

$$\mathcal{L}\left(f(t)\right) = \int_0^\infty f(t)e^{-st}dt \tag{06.5}$$

which leads directly to the familiar Laplace transform properties (1) of linearity and (2) of differentiation: the Laplace transform of the derivative of a function $f(t)$ (with zero initial conditions) is s times the transform of the function $F(s) \equiv \mathcal{L}(f(t))$:

$$\mathcal{L}\left(\frac{df(t)}{dt}\right) = sF(s). \tag{06.6}$$

06.03.2 Continuous transfer functions

continuous transfer function

These properties allow us to find the transfer function of a linear *continuous* system, given its *differential* equation. We define the *continuous transfer function* $T(s)$ to be the Laplace transform of the output $Y(s)$ divided by the Laplace transform of the input $X(s)$; i.e.

$$T(s) = \frac{Y(s)}{X(s)}. \tag{06.7}$$

Reconsider the continuous differential equation for a dynamic system Equation 06.1. The equivalent transfer function, using the linearity and differentiation properties of the Laplace transform, is

$$T(s) = \frac{\beta_m s^m + \beta_{m-1} s^{m-1} + \cdots + \beta_1 s^1 + \beta_0}{\alpha_n s^n + \alpha_{n-1} s^{n-1} + \cdots + \alpha_1 s^1_+ \alpha_0} \tag{06.8}$$

where α_k and β_k are the same constants that appeared in Equation 06.1.

06.03.3 *z*-Transforms

z-transform

For *discrete* systems and their *difference* equations, a very similar procedure is available. The *z-transform* $F(z) \equiv \mathcal{Z}(f(n))$ of a sequence $f(n)$, with complex variable z (analogous to s), is defined by[3]

$$\mathcal{Z}(f(n)) = \sum_{n=0}^{\infty} f(n) z^{-n}. \tag{06.9}$$

This leads directly to the z-transform properties (1) of linearity and (2) of delay, analogous to (06.6) for discrete systems: the z-transform of a function delayed by one sample period is z^{-1} times the transform of the function $F(z)$:

$$\mathcal{Z}(f(n-1)) = z^{-1} F(z), \tag{06.10}$$

06.03.4 Discrete transfer functions

discrete transfer function

We define the *discrete transfer function* $T(z)$ to be the z-transform of the output $Y(z)$ divided by the z-transform of the input $X(z)$; i.e.

$$T(z) = \frac{Y(z)}{X(z)}. \tag{06.11}$$

Given the z-transform properties, we can easily find the transfer function of a *discrete* system given its *difference* equation.

Example 06.03-1 discrete transfer function

What is the discrete tranfer function corresponding to the second-order difference equation

$$\begin{aligned} &a_0 y(n) + a_1 y(n-1) + a_2 y(n-2) = \\ &= b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) \end{aligned} \tag{06.12}$$

with constants a_n and b_n?

The z-transform of the difference equation is determined by linearity and successively applying (06.10) to arrive at

$$\left(1 + a_1 z^{-1} + a_2 z^{-2}\right) Y(z) = \left(b_0 + b_1 z^{-1} + b_2 z^{-2}\right) X(z). \tag{06.13}$$

[3] There are many more uses for z-transforms. For more details, see Franklin et al. (1998).

Rearranging, the discrete transfer function is

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{06.14}$$

Notice that the transfer function (06.14) and the difference equation (06.12), can be derived from each other by inspection. Notice also that the transfer function of a discrete system is the ratio of two polynomials in z, just as the transfer function of a continuous system is the ratio of two polynomials in s.

06.03.5 Discrete approximations of continuous transfer functions

There are several ways to derive an approximate discrete transfer function from a corresponding continuous transfer function. We will use a popular technique called *Tustin's method* that approximates a continuous function of time by straight lines connecting the sampled points (i.e. trapezoidal integration).

Tustin's method

The discrete transfer function is found using Tustin's method by making the following substitution:

$$s \mapsto \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \tag{06.15}$$

and rewriting the transfer function in the form of equation (06.14). Here, T is the sample period.

Example 06.03-2 Tustin's method

Consider a continuous first order system described by the transfer function:

$$\frac{Y(s)}{X(s)} = \frac{1}{\tau s + 1}, \text{ where } \tau \text{ is the time constant.} \tag{06.16}$$

Using Tustin's method, derive a discrete transfer function and the corresponding difference equation.

Substituting Equation 06.15 into the transfer function, we have:

$$\frac{Y(z)}{X(z)} = \frac{\alpha + \alpha z^{-1}}{1 - (1 - 2\alpha) z^{-1}},$$

where α is a constant:

$$\alpha = \frac{T}{2\tau + T}$$

from which the difference equation can be inferred (see Equations 06.12 to 06.14 above):

$$y(n) = (1 - 2\alpha)y(n-1) + \alpha x(n) + \alpha x(n-1)$$

Notice again that the current value of the output $y(n)$ depends on the previous output, $y(n-1)$, and on the *current* and previous inputs, $x(n)$ and $x(n-1)$.
Notice also that the coefficients depend on the time constant τ in the original continuous system and on the sample period T.
During each sample period, the value of the current value of the input $x(n)$ is measured and the current value of the output $y(n)$ is computed. Suppose that the time constant $\tau = 2$, the sample period $T = 1$, and that the input is a unit step ($x(n) = 1$ for all n), and the initial condition $y(0) = 0$.
Then, from our solution for $y(n)$,

$$y(n) = 0.6y(n-1) + 0.4 \tag{06.17}$$

and we can compute the output sequence:

Figure 06.3 shows plots of the input and output sequences.

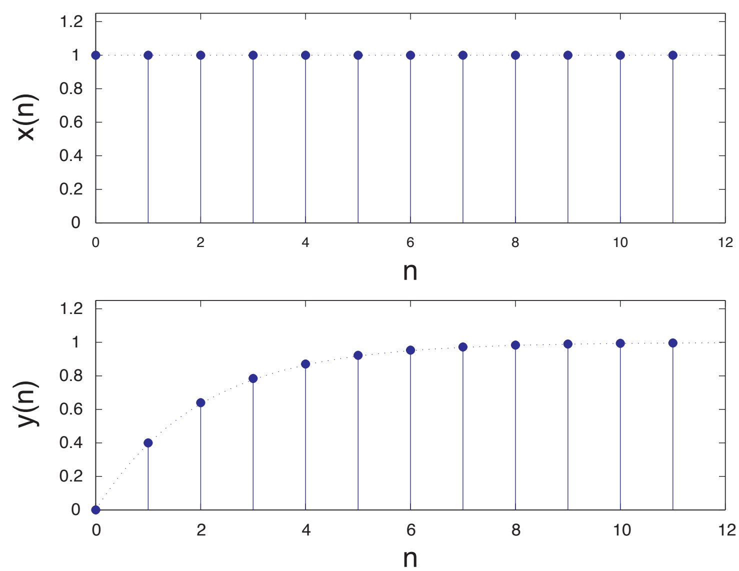

Figure 06.3: input and output sequences.

The dotted line is the exact solution $y(t/T)$ of the original continuous *differential* equation. As you can see, in this example, Tustin's method is very close to the exact solution at the sample points.

See Resource 13 for a table of common controller transfer functions converted to discrete transfer functions via Tustin's method.

06.03.6 Matlab's `c2d`

The Matlab Control Systems Toolbox includes a function `c2d` that computes the Tustin equivalent discrete system `sysd` from the continuous system `sys`, as follows.

```
sysd = c2d(sys, T, 'tustin')
```

This function can also use other common techniques to yield a discrete approximation of a continuous transfer function.