
Digital Controller Design and Simulation

Digital controllers are far more convenient to implement on microprocessors than are continuous-time controllers. Given a continuous-time controller, designed by any technique (state-space methods, root locus, PID/lead/lag, etc.), one may convert it to a digital controller in several ways. One effective method is to employ the bilinear transformation, as described here. First, the Z-transform and difference equations are briefly covered.

Z-Transform

Given a time series $y_k u_{-1}(k) = y_0, y_1, y_2, \dots$, its Z-transform is given by

$$Y(z) = \sum_{k=0}^{\infty} y_k z^{-k}.$$

The discrete time variable is k . The one-sided transform is used (i.e. lower limit of zero) since the time series is multiplied by the discrete unit step $u_{-1}(k)$. The Z-transform variable z is a complex variable, like the Laplace transform variable $s = \sigma + j\omega$.

In connection with finding Z-transforms it is useful to recall the series identity

$$\sum_{k=0}^{N-1} a^k = \frac{1 - a^N}{1 - a}.$$

If $|a| < 1$, the series converges and one has also

$$\sum_{k=0}^{\infty} a^k = \frac{1}{1 - a}.$$

Example. The Z-transform of the discrete-time exponential $y_k = a^k u_{-1}(k) = a^0, a^1, a^2, \dots$ is given by

$$Y(z) = \sum_{k=0}^{\infty} a^k z^{-k} = \sum_{k=0}^{\infty} (az^{-1})^k = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}.$$

This has a pole at $z=a$ and a zero at $z=0$.

Example. The discrete unit step $u_{-1}(k) = 1, 1, 1, \dots$ (i.e. a sequence of 1's which begins at time $k=0$) is a special case of the discrete exponential having $a=1$. Its Z-transform is

$$Y(z) = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1},$$

which has a pole at $z=1$ and a zero at $z=0$.

Example. The discrete unit pulse $u_0(k) = 1, 0, 0, \dots$ (i.e. a one occurring at time $k=0$) has transform

$$Y(z) = \sum_{k=0}^{\infty} u_0(k) = 1,$$

i.e. a constant surface which has no poles or zeros.

Note that Z-transforms of causal signals beginning at $k=0$ generally have a z in the numerator. This is due to the fact that what appears in the summation is z^{-1} .

Table of Z Transforms

Unit pulse 1,0,0,...		1
Unit step $u_{-1}(k) = 1, 1, 1, \dots$		$\frac{1}{1 - z^{-1}}$
Discrete ramp $k = 0, 1, 2, 3, \dots$		$\frac{z^{-1}}{(1 - z^{-1})^2}$
Discrete exponential a^k		$\frac{1}{1 - az^{-1}}$
Sampled Signals, $T =$ sampling period		
$\sin \omega t$	$\sin \omega kT$	$\frac{z^{-1} \sin \omega T}{1 - 2z^{-1} \cos \omega T + z^{-2}}$
$\cos \omega t$	$\cos \omega kT$	$\frac{1 - z^{-1} \cos \omega T}{1 - 2z^{-1} \cos \omega T + z^{-2}}$
$e^{-\alpha t} \sin \omega t$	$e^{-\alpha kT} \sin \omega kT$	$\frac{e^{-\alpha T} z^{-1} \sin \omega T}{1 - 2e^{-\alpha T} z^{-1} \cos \omega T + e^{-2\alpha T} z^{-2}}$
$e^{-\alpha t} \cos \omega t$	$e^{-\alpha kT} \cos \omega kT$	$\frac{1 - e^{-\alpha T} z^{-1} \cos \omega T}{1 - 2e^{-\alpha T} z^{-1} \cos \omega T + e^{-2\alpha T} z^{-2}}$

Sampling of Continuous-Time Signals

The continuous-time exponential is $y(t) = e^{\alpha t} u_{-t}(t)$, where for stability the real part of the pole α is negative. Selecting a SAMPLING PERIOD T , one relates the continuous and discrete time variables by

$$t = kT.$$

Then, the continuous exponential becomes

$$e^{\alpha t} = e^{akT} = (e^{aT})^k = a^k$$

with $a = e^{aT}$. This provides a mapping between continuous poles and discrete poles of time functions.

For instance, if the continuous-time pole is at $s = \alpha = -2$ and the sampling period is $T = 10\text{msec} = 0.01$ sec, then one has the discrete pole at $z = e^{-2(0.01)} = e^{-0.02} = 0.98$. Note that a continuous pole at $s = 0$ maps to a discrete pole at $z = e^0 = 1$.

Given this mapping, it is now instructive to compare the examples above with their continuous-time counterparts. In fact, note that the continuous unit step, which has the Laplace transform of $1/s$, has a pole at $s=0$, while the discrete unit step has a pole at $z=1$. The continuous exponential $e^{\alpha t}$ has a pole at $s=\alpha$, while the discrete exponential has a pole at $z = e^{aT}$, with T the sampling period. The discrete unit pulse and the (continuous) unit impulse both have constant transforms of 1.

Inverse Z-Transform

The inverse Z-transform may be found in several ways.

Example. Inverse Z-Transform by Long Division

Suppose $Y(z) = \frac{z}{z^2 - 1.7z + 0.72}$. One can determine the associated time series y_k by writing this in the form

$$Y(z) = \sum_{k=0}^{\infty} y_k z^{-k} = y_0 + y_1 z^{-1} + y_2 z^{-2} + \dots$$

by any technique. One way to do this is by long division. In fact, one may write

$$\begin{array}{r} z^2 - 1.7z + 0.72 \overline{) z^{-1} + 1.7z^{-2} + \dots} \\ \underline{z - 1.7 + 0.72z^{-1}} \\ 1.7 - 0.72z^{-1} \\ \underline{1.7 - 2.89z^{-1} + 1.224z^{-2}} \end{array}$$

The inverse Z-transform appears in the quotient, so that $y_k = 0, 1, 1.7, \dots$

Example. Inverse Z-Transform by Partial Fraction Expansion

Exactly as for the Laplace transform inverse, one may use the PFE. Thus, write

$$Y(z) = \frac{z}{z^2 - 1.7z + 0.72} = \frac{1}{(z - 0.9)(z - 0.8)} \cdot z = \left[\frac{10}{z - 0.9} + \frac{-10}{z - 0.8} \right] \cdot z = \frac{10z}{z - 0.9} + \frac{-10z}{z - 0.8}.$$

The residues are determined exactly as in the continuous-time case. Note, however, that one keeps the factor z out of the PFE to obtain the z in the numerators of the expansion. This is necessary since Z-transforms generally have a zero at $z=0$.

Now, according to the examples worked above, the time series is given as

$$y_k = 10(0.9^k - 0.8^k)u_{-1}(k).$$

Evaluating the first few terms gives $y_k = 0, 1, 1.7, \dots$. This is the same result obtained by long division.

The PFE is generally preferable to long division since it yields closed-form solutions.

Difference Equations

To simulate the continuous-time state equation $\dot{x} = f(x, u)$ one must use a numerical integrator such as Runge-Kutta. This means that to implement a controller described in terms of differential equations, one needs a numerical integrator, which is inconvenient. This applies to any controller $K(s)$ described in terms of Laplace transforms. The simulation of DIFFERENCE EQUATIONS is far easier than the simulation of differential equations. This means that controllers described in terms of difference equations are very easy to implement on a digital microprocessor. This applies to controllers $K(z)$ described in terms of Z-transforms. Before showing how to design digital controllers, we briefly discuss difference equations, and relate them to Z-transforms.

A linear time-invariant system may be described in discrete time in terms of its DISCRETE TRANSFER FUNCTION $H(z)$. Then the input $U(z)$ and output $Y(z)$ are related by

$$Y(z) = H(z)U(z).$$

Writing more detail in terms of the numerator and denominator polynomials, one has

$$Y(z) = H(z)U(z) = \frac{b_0 z^m + b_1 z^{m-1} + \dots + b_{m-1} z + b_m}{z^n + a_1 z^{n-1} + \dots + a_{n-1} z + a_n} U(z)$$

where the denominator degree is n , the numerator degree is $m \leq n$, and the relative degree is $n-m$.

Dividing through by z^n one may write

$$Y(z) = H(z)U(z) = z^{-d} \frac{b_0 + b_1 z^{-1} + \cdots + b_{m-1} z^{-(m-1)} + b_m z^{-m}}{1 + a_1 z^{-1} + \cdots + a_{n-1} z^{-(n-1)} + a_n z^{-n}} U(z)$$

where the SYSTEM DELAY is $d = n - m$. Now one writes

$$(1 + a_1 z^{-1} + \cdots + a_n z^{-n})Y(z) = z^{-d} (b_0 + b_1 z^{-1} + \cdots + b_m z^{-m})U(z).$$

We will abuse notation now and define an operator z^{-1} in the time domain. To see what $z^{-1}y_k$ means, write its transform as

$$Y'(z) = \sum_{k=0}^{\infty} (z^{-1}y_k)z^{-k} = \sum_{k=0}^{\infty} y_k z^{-(k+1)}$$

Now change variables to $K = k + 1$ to see that

$$Y'(z) = \sum_{K=1}^{\infty} y_{K-1} z^{-K}.$$

That is, $z^{-1}y_k$ is a delayed version y_{k-1} of the time series.

The Z-transform SHIFT PROPERTY says that multiplying the Z-transform by z^{-1} delays the time series by one. Compare this to the Laplace transform property which says that multiplying the transform by $1/s$ amounts to integrating the time function.

In continuous-time systems, the memory resides in the integrators $1/s$. In discrete-time systems, the memory resides in the delays z^{-1} . A series of delays is nothing but a shift register.

Now one may write the input/output relation of the digital system as

$$y_k + a_1 y_{k-1} + \cdots + a_n y_{k-n} = b_0 u_{k-d} + b_1 u_{k-d-1} + \cdots + b_m u_{k-d-m}$$

or

$$y_k = -a_1 y_{k-1} - \cdots - a_n y_{k-n} + b_0 u_{k-d} + b_1 u_{k-d-1} + \cdots + b_m u_{k-d-m}.$$

This is a difference equation describing the digital system.

Example. Let the digital transfer function be

$$H(z) = \frac{z}{z^2 - 1.7z + 0.72} = z^{-1} \frac{1}{1 - 1.7z^{-1} + 0.72z^{-2}}.$$

Then one has $Y(z) = H(z)U(z)$ or

$$(1 - 1.7z^{-1} + 0.72z^{-2})Y(z) = z^{-1}U(z)$$

so that in the time domain one has the difference equation

$$y_k = 1.7y_{k-1} - 0.72y_{k-2} + u_{k-1}.$$

The system delay is $d = 1$.

The meaning of the system delay is that a control u_k applied at time k has no influence on the output until time $k + d$. In the, last example, for instance, u_k affects y_{k+1} , not y_k .

Discretization of Continuous-Time Controllers

By any of a variety of techniques, one may design a continuous-time compensator $K(s)$. This may be converted to digital form $K(z)$ using several techniques, among the most direct of which is the bilinear transformation (BLT).

The relation between the Laplace transform variable s and the Z-transform variable z is $z=e^{sT}$, with T the sampling period. However, using this to transform $K(s)$ to $K(z)$ will give non-polynomial transfer functions. Note that

$$e^{sT} \approx \frac{1 + sT/2}{1 - sT/2}$$

Therefore define the BLT by

$$z = \frac{1 + sT/2}{1 - sT/2},$$

and its inverse

$$s = \frac{2}{T} \frac{z-1}{z+1}.$$

To convert a continuous transfer function $K(s)$ to a discrete transfer function using sample period T , then, one simply replaces all occurrences of s by $\frac{2}{T} \frac{z-1}{z+1}$. That is

$$K(z) = K(s) \Big|_{s=\frac{2}{T} \frac{z-1}{z+1}}.$$

Conversion of LPF to Digital Form

A continuous-time low pass filter is given by

$$K(s) = \frac{\alpha}{s + \alpha}.$$

This could be a compensation network designed using, e.g. root locus techniques. To convert this to a digital controller one writes

$$K(z) = \frac{\alpha}{\left(\frac{2}{T} \frac{z-1}{z+1}\right) + \alpha} = \frac{z+1}{z-a},$$

where the digital filter pole is at $a = \frac{1 - \alpha T/2}{1 + \alpha T/2}$. Note that the continuous filter has a pole at $s = -\alpha$.

Note that here the BLT yields a zero at $z=-1$. This is because there is one zero at infinity in $K(s)$. The BLT maps finite poles and zeros according to

$$z = \frac{1 + sT/2}{1 - sT/2}.$$

Digital PID Controller

The continuous-time PID controller can be written in the form

$$K(s) = k \left[1 + \frac{1}{T_i s} + \frac{T_d s}{1 + T_d s / \eta} \right]$$

where T_i is the integration time constant or 'reset time', T_d is the derivative time constant, and η is a large filtering pole (usually set to about 10 by the manufacturer). Unfortunately, to implement this PID controller, one would require some sort of numerical integration routine such as Runge-Kutta.

To convert this to digital form using the BLT, write

$$K(z) = k \left[1 + \frac{1}{T_i \left(\frac{2}{T} \frac{z-1}{z+1} \right)} + \frac{T_d \left(\frac{2}{T} \frac{z-1}{z+1} \right)}{1 + \frac{T_d}{\eta} \left(\frac{2}{T} \frac{z-1}{z+1} \right)} \right].$$

This may be simplified to obtain

$$K(z) = k \left[1 + \frac{T}{T_{iD}} \frac{z+1}{z-1} + \frac{T_{dD}}{T} \frac{z-1}{z-\eta_D} \right],$$

where the digital integral and derivative time constants are

$$T_{iD} = 2T_i$$

$$T_{dD} = \frac{\eta T}{1 + \eta T / 2T_d}$$

and the discrete filtering pole is given by

$$\eta_D = \frac{1 - \eta T / 2T_d}{1 + \eta T / 2T_d}.$$

It is easy to implement this digital PID controller using difference equations. In fact, if the control input is given by $u_k = K(z)e_k$, with e_k the tracking error, then divide through by the highest power of z to obtain

$$u_k = K(z^{-1})e_k = k \left[1 + \frac{T}{T_{iD}} \frac{1+z^{-1}}{1-z^{-1}} + \frac{T_{dD}}{T} \frac{1-z^{-1}}{1-\eta_D z^{-1}} \right] e_k.$$

Now implement the I term separately as

$$u_k^i = \frac{T}{T_{iD}} \frac{1+z^{-1}}{1-z^{-1}} e_k$$

which yields

$$u_k^i = u_{k-1}^i + \frac{T}{T_{iD}} (e_k + e_{k-1}).$$

Implement the D term as

$$u_k^d = \frac{T_{dD}}{T} \frac{1-z^{-1}}{1-\eta_D z^{-1}} e_k$$

or

$$u_k^d = \eta_D u_{k-1}^d + \frac{T_{dD}}{T} (e_k - e_{k-1}).$$

The complete control input is then given by the equation set

$$u_k^i = u_{k-1}^i + \frac{T}{T_{iD}} (e_k + e_{k-1}).$$

$$u_k^d = \eta_D u_{k-1}^d + \frac{T_{dD}}{T} (e_k - e_{k-1})$$

$$u_k = k(e_k + u_k^i + u_k^d).$$

This is the digital controller. No Runge-Kutta routine is needed to implement it, only difference equations, which are easily programmed on a computer.

Note that the integral input is computed by adding to its previous value the average of two consecutive error terms. The derivative term is computed from the difference between two consecutive error terms.