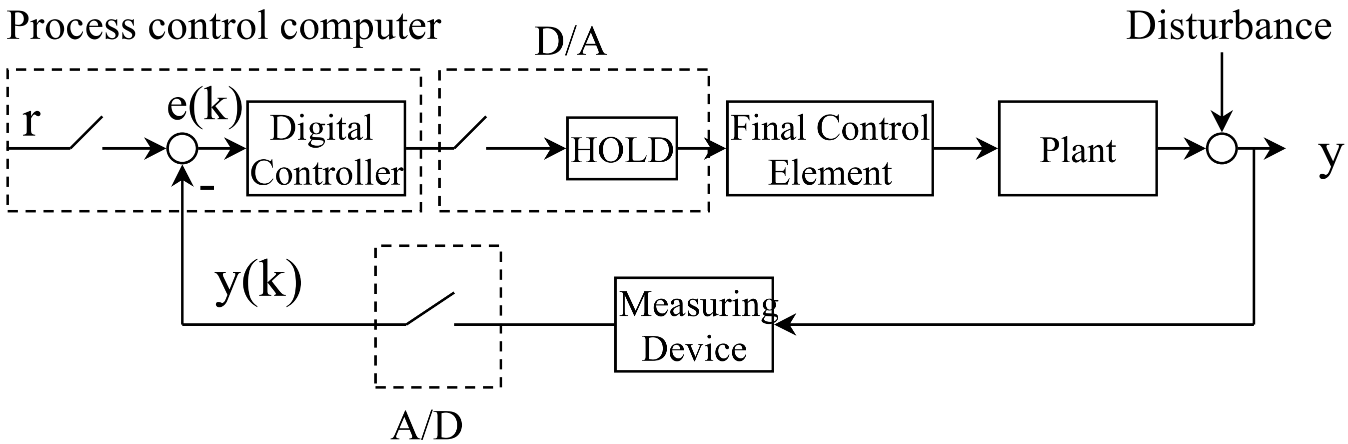
CHE 576: Intermediate Process Control
LECTURE NOTES
(part 1)

Biao Huang
Department of Chemical and Material Engineering
University of Alberta

1. Introduction to Discrete-time Systems



A/D: convert analog signal to digital or discrete signal
D/A: convert digital or discrete signal to analog signal
Hold: convert discrete signal to piece-wise continuous signal
Digital Controller: performs calculation according to a specified control law

The advantages of using digital computer as the controller:

- Data acquisition from the physical process is very easily and rapidly done with the digital computer.
- A tremendous capacity for mass storage (and rapid retrieval) of the collected process information is readily available.
- Complicated control law can be easily implemented in a digital computer.
- The required computations, no matter how complex, can be carried out at very high speeds.
- The cost of digital computers and ancillary equipment has reduced drastically in the last few years.

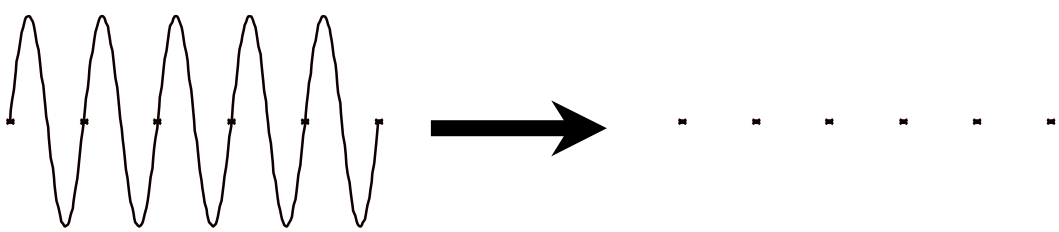
What are the specialties of a computer control system:

- Continuous signal is discretized. We may lose some information of the signal.
- A computer can only give out information in digital form at discrete points in time. A hold-device must be used to “hold” the previous value of the digital signal until another sample is available.
- A control law is performed in the digital or difference equation form.

Selection of the sampling period:

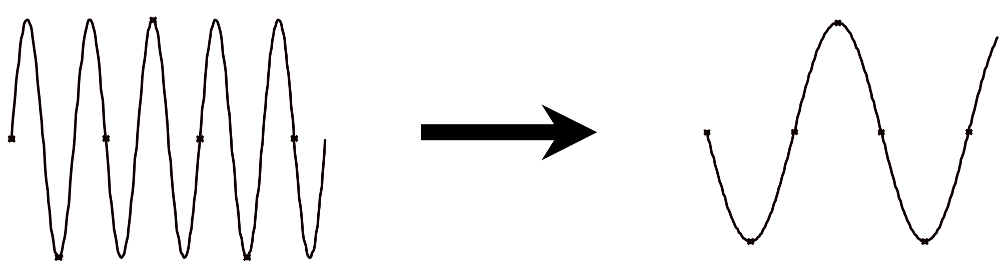
As discussed in the previous slide, when a digital computer is used for control, continuous measurements are converted into digital form by an analog to digital converter. This operation is necessary because the digital computer cannot directly process an analog signal. The signal must be sampled at discrete points in time. The time interval between successive samples is referred to as the sampling period

The sampling period must be small enough so that significant process information is not lost.



One sample per cycle

The information is lost



4/3 sample per cycle

The signal cannot be exactly recovered

These phenomena are known as aliasing.

Time constant: τ

For a first order system, the time elapsed for process response to reach 63.2% of its final value is defined as the time constant.
The second or higher order overdamped system can be approximated by a first order system with time delay.

Choose sampling period for the first or higher order overdamped system:
Practical experience suggests that a sampling period between 0.1 and 0.2 of one time constant yields satisfactory results. If there is time delay τ_d and the time delay has same order of magnitude as the time constant, select the sampling period equal to $0.1\tau - 0.2\tau$ or $0.1\tau_d - 0.2\tau_d$ whichever is smaller. If τ_d is much smaller than τ, neglect the time delay and take the sampling period as $0.1\tau - 0.2\tau$.

Choose sampling period for the oscillating response of a system:
Oscillatory behavior is exhibited by underdamped open- or closed-loop systems and by the steady-state response of linear systems subject to periodic, sinusoidal input changes. To develop a good sampled representation of an oscillating signal, follow the rule:

Sample an oscillating signal more than two times per cycle of oscillation; otherwise, it is impossible to reconstruct the original signal from its sampled values as discussed from the previous slide.

Zero-order Hold
The discrete nature of a digital computer implies that when a computer is used to control a process, the controller output is a sequence of discrete signals or impulses. Thus a valve opens when a control impulse from the computer reaches the valve, but then it closes until the next control impulse arrives at the valve. Such control action is undesirable. A zero-order hold is a device that can hold the impulse at a constant until the next impulse arrives.

2. Intro. to Discrete-time Systems - contd.

Conversion of Continuous- to Discrete-time Model (Seborg et al.)

FINITE DIFFERENCE MODELS

A digital computer by its very nature deals internally with discrete data or numerical values of functions. To perform analytical operations such as differentiation and integration, numerical approximations must be utilized. One way of converting continuous-time models to discrete-time form is to use finite difference techniques. In general, a differential equation,

$$\frac{dy(t)}{dt} = f(y, x) \tag{1}$$

where y is the output variable and x is the input variable, can be numerically integrated (although with some error) by introducing a finite difference approximation for the derivative. For example, the first-order, backward difference approximation to the derivative at time $t = n\Delta t$ is

$$\frac{dy}{dt} \approx \frac{y_n - y_{n-1}}{\Delta t} \tag{2}$$

where Δt is the integration interval that is specified by the user, y_n is the value of y_t at $t = n\Delta t$ and y_{n-1} denotes the value at the previous sampling instant $t = (n-1)\Delta t$. Substituting (2) into (1) and evaluating function $f(y, x)$ at the previous values of y and x (i.e., y_{n-1} and x_{n-1}) gives

$$\frac{y_n - y_{n-1}}{\Delta t} \approx f(y_{n-1}, x_{n-1}) \tag{3}$$

or

$$y_n = y_{n-1} + \Delta t f(y_{n-1}, x_{n-1}) \tag{4}$$

Equation (4) is a first-order difference equation that can be used to predict the value of y at time step n based on information at the previous time step $(n-1)$, namely, y_{n-1} and $f(y_{n-1}, x_{n-1})$. This type of expression is called a recurrence relation. It can be used to numerically integrate Equation (1) by calculating y_n for $n = 0, 1, 2, \cdots$ starting from known initial conditions, $y(0)$ and $x(0)$. In general, the resulting numerical solution, $\{y_n, n = 1, 2, 3, \cdots\}$ becomes more accurate and approaches the correct solution $y(t)$ as Δt decreases. However, for extremely small values of Δt, computer roundoff errors can be a significant source of error.

EXACT DISCRETIZATION FOR LINEAR SYSTEMS

For a process described by a linear differential equation, an alternative discrete-time model can be derived based on the analytical solution for a piecewise constant input. This approach yields an exact discrete-time model if the input variable is actually constant between sampling instants. Thus, this analytical approach eliminates the discretization error inherent in finite difference approximations for the important practical situation where the digital computer output (process input) is held constant between sampling instants. This signal is piecewise constant if the digital to analog device acts as a zero-order hold.

Example: Consider a first-order differential equation:

$$\frac{dy(t)}{dt} = \frac{1}{\tau}y(t) + \frac{1}{\tau}x(t) \tag{5}$$

Assuming $x(t)$ is piecewise continuous signal, use the exact discretization method to find the discrete-time model. Assuming that $x(t)$ is constant such that $x(t) = x(0)$, and that $y(0) \neq 0$. Taking the Laplace transform of equation (5) gives

$$sY(s) - y(0) = -\frac{1}{\tau}Y(s) + \frac{1}{\tau}\frac{x(0)}{s}$$

Solving for $Y(s)$,

$$Y(s) = \frac{1}{s + 1/\tau}[\frac{1}{\tau}\frac{x(0)}{s} + y(0)]$$

and taking inverse Laplace transform gives

$$y(t) = x(0)(1 - e^{-t/\tau}) + y(0)e^{-t/\tau} \tag{6}$$

Equation (6) is valid for all values of t. Thus, after one sampling period, $t = \Delta t$, and we have

$$y(\Delta t) = x(0)(1 - e^{-\Delta t/\tau}) + y(0)e^{-\Delta t/\tau} \tag{7}$$

Next, we can generalize the analysis by considering the time interval, $(n-1)\Delta t$ to $n\Delta t$. For an initial condition $y[(n-1)\Delta t]$ and a constant input, $x(t) = x[(n-1)\Delta t]$ for $(n-1)\Delta t \leq t < n\Delta t$, the analytical solution to (7) is

$$y(n\Delta t) = x[(n-1)\Delta t](1 - e^{-\Delta t/\tau}) + y[(n-1)\Delta t]e^{-\Delta t/\tau} \tag{8}$$

Note that the exponential terms are the same as for Equation (7). Finally, equation (8) can be written more compactly as

$$y_n = e^{-\Delta t/\tau}y_{n-1} + (1 - e^{-\Delta t/\tau})x_{n-1}$$

USING MATLAB FUNCTIONS

The Matlab function *c2d* can be used for exact discretization. Use help function to find more detailed description on this function.

Consider a first order differential equation:

$$\frac{dy}{dt} = -0.5y + 0.5u$$

The time constant of this first order differential equation is $\tau = 2$. If the sampling period $\Delta t = 0.2$, then the exact discretization yields

$$y_n = e^{-\Delta t/\tau} y_{n-1} + (1 - e^{-\Delta t/\tau}) u_{n-1} = 0.9048 y_{n-1} + 0.0952 u_{n-1}$$

In order to use Matlab function, we have to transfer the differential equation into the transfer function which is

$$\frac{Y(s)}{U(s)} = \frac{1}{2s+1}$$

Thus $num = 1$ and $den = [2, 1]$, and then use $sysc = tf(num, den)$ to generate a continuous transfer function, followed by $sysd = c2d(sysc, 0.2)$. This yields a discrete transfer function $sysd$ that has numerator and denominator of $numd = [0, 0.0952]$ and $dend = [1, -0.9048]$, The discrete transfer function will be discussed in the next section. For the time being, we can consider the numerator $numd$ contains the coefficients of u_n, u_{n-1} and the denominator $dend$ contains the coefficients of y_n, y_{n-1}. Therefore the difference equation calculated from Matlab function is

$$y_n - 0.9048 y_{n-1} = 0.0952 u_{n-1}$$

which is the same as the hand calculation.

3. Introduction to Discrete-time Systems - Contd.

THE BACKSHIFT OPERATOR AND TRANSFER FUNCTION

The backshift operator q^{-1} is an operator which moves a signal one step back, i.e. $q^{-1}y_n = y_{n-1}$. Similarly, $q^{-2}y_n = q^{-1}y_{n-1} = y_{n-2}$ and $qy_n = y_{n+1}$. It is convenient to use the backshift operator to write a difference equation. For example, a difference equation

$$y_n = 1.5y_{n-1} - 0.5y_{n-2} + 0.5u_{n-1} \tag{1}$$

can be represented as

$$y_n = 1.5q^{-1}y_n - 0.5q^{-2}y_n + 0.5q^{-1}u_n$$

This can be written as

$$\frac{y_n}{u_n} = \frac{0.5q^{-1}}{1 - 1.5q^{-1} + 0.5q^{-2}} \tag{2}$$

This equation is also regarded as a discrete transfer function. Time-delays can easily be represented by the backshift operator. For example, e^{-5s} represents 5 units of time-delay for a continuous-time system. If the sampling period is 1 unit, then this delay can be represented as q^{-5} for the discrete-time system.

Z-transform is often used as a transformation for discrete-time systems. The operator Z^{-1} plays the same role as the backshift operator q^{-1}. Therefore, we will not distinguish them in this course. For example, equation (2) may be written as

$$\frac{y_n}{u_n} = \frac{0.5Z^{-1}}{1 - 1.5Z^{-1} + 0.5Z^{-2}}$$

or equivalently

$$\frac{y_n}{u_n} = \frac{0.5Z}{Z^2 - 1.5Z + 0.5}$$

DISCRETE-TIME RESPONSE VIA DIRECT SUBSTITUTION

Suppose that in equation (1), the discrete input u_t takes values as

$$u_0 = 1, u_1 = 0, u_2 = 0, u_3 = 0 \cdots$$

i.e.

$$u_n = \begin{cases} 1 & n = 0 \\ 0 & n > 0 \end{cases}$$

This input signal is also regarded as the impulse. The response y_n with this impulse input is therefore called as impulse response. Suppose initial value of y_n is zero, i.e. $y_n = 0$ for $n \leq 0$, then the impulse response of y_n may be directly calculated from equation (1) as

$$\begin{aligned} y_1 &= 1.5y_0 - 0.5y_{-1} + 0.5u_0 = 0.5 \\ y_2 &= 1.5y_1 - 0.5y_0 + 0.5u_1 = 0.75 \\ y_3 &= 1.5y_2 - 0.5y_1 + 0.5u_2 = 0.875 \\ &\vdots \end{aligned}$$

DISCRETE-TIME RESPONSE VIA LONG DIVISION

Alternatively, if we expend equation (2) using long division, we will get an infinite series as

$$\frac{y_n}{u_n} = 0.5q^{-1} + 0.75q^{-2} + 0.875q^{-3} + \cdots$$

i.e., the coefficients of this infinite series are impulse response of y_n. This provides an alternative way to solve the difference equation (1).

DISCRETE-TIME RESPONSE VIA PARTIAL FRACTION EXPANSION

The infinite series expansion of a transfer function can be calculated via long division. Some simple transfer functions can easily be expended into the infinite series without actually carrying the long division. For example, it is well-known that

$$\frac{r}{1 - pq^{-1}} = r + rpq^{-1} + rp^2q^{-2} + \cdots \tag{3}$$

Therefore, the impulse response for a difference equation expressed in the backshift operator

$$y_n = \frac{r}{1 - pq^{-1}} u_t$$

will be

$$y_0 = r, y_1 = rp, y_2 = rp^2, \cdots$$

or simply

$$y_n = rp^n \tag{4}$$

The transfer function (3) is also regarded as the first-order discrete transfer function. Now consider a higher order transfer function:

$$G(q^{-1}) \triangleq \frac{y_n}{u_n} = \frac{v(q^{-1})}{(1-p_1q^{-1})(1-p_2q^{-1})\cdots(1-p_mq^{-1})}$$

Using partial fraction expansion, we can expend this higher order transfer function into the summation of first order transfer functions:

$$G(q^{-1}) = \frac{r_1}{1-p_1q^{-1}} + \frac{r_2}{1-p_2q^{-1}} + \cdots + \frac{r_m}{1-p_mq^{-1}}$$

Then according to equation (4), the impulse response y_n is

$$y_n = r_1(p_1)^n + r_2(p_2)^n + \cdots + r_m(p_m)^n$$

The Matlab function *impulse* can be used to generate the impulse response of a discrete transfer function. Use help function to see more detailed description on this function. For example, in equation 2, we can set $numd = [0, 0.5, 0]$ and $dend = [1, -1.5, 0.5]$, where the numerator *numd* contains the coefficients of the numerator of the discrete transfer function i.e., q^0, q^{-1}, q^{-2}, and *dend* contains the coefficients of the denominator of the discrete transfer function i.e., q^0, q^{-1}, q^{-2}. When we specify the numerator and the denominator for a discrete transfer function, we have to keep the same length of the numerator and denominator. If they do not have the same length, add zeros to the right of the row vectors in order to make them have the same length. Also the first coefficient must begin with q^0. For example, consider a discrete transfer function

$$G(q^{-1}) = \frac{1-q^{-1}}{1-q^{-1}+q^{-2}}$$

then its denominator is $den = [1, -1, 1]$ and its numerator is $num = [1, -1, 0]$ (add zeros to keep the same length as the denominator). If the discrete transfer function is

$$G(q^{-1}) = \frac{q^{-1} - 0.5q^{-2}}{1 - q^{-1} + q^{-2}}$$

then $den = [1, -1, 1]$ and $num = [0, 1, -0.5]$ (always begins with the coefficient of q^0).

DISCRETE-TIME RESPONSE OF TRANSFER FUNCTIONS:

As discussed in the previous lecture, the impulse response of a first order discrete transfer function

$$\frac{y_n}{u_n} = G_{yu}(q^{-1}) = \frac{r}{1 - pq^{-1}}$$

is $y_n = rp^n$ (i.e., if u_n is an impulse). On the other hand, given a signal $u_n = rp^n$, we may think that it is generated from an impulse response of a discrete transfer function

$$\frac{u_n}{\delta_n} = G_{u\delta}(q^{-1}) = \frac{r}{1 - pq^{-1}}$$

where δ_n is an impulse.

In general, if $y_n = G_{yu}(q^{-1})u_n$ and $u_n = G_{u\delta}(q^{-1})\delta_n$ (where δ_n is an impulse), then the response of y_n with respect to the arbitrary signal $u_n = G_{u\delta}(q^{-1})\delta_n$ will be

$$y_n = G_{yu}(q^{-1})u_n = G_{yu}(q^{-1})G_{u\delta}(q^{-1})\delta_n$$

This can be written as a transfer function form as

$$\frac{y_n}{\delta_n} = G_{y\delta}(q^{-1}) = G_{yu}(q^{-1})G_{u\delta}(q^{-1})$$

Therefore, the response of y_n to an arbitrary signal u_n via the transfer function (or difference equation) $G_{yu}(q^{-1})$ is equivalent to the impulse response of y_n via the transfer function $G_{y\delta}(q^{-1})$.

Example: Consider a discrete signal $u_n = 1$ for $n \geq 0$ and $u_n = 0$ for $n < 0$. This signal is regarded as a step signal and can be generated from the impulse response of the following discrete transfer function:

$$\frac{u_n}{\delta_n} = \frac{1}{1 - q^{-1}}$$

where δ_n is an impulse.

Example: Calculate the step response of the first order discrete transfer function.

$$\begin{aligned} y_n &= G_{yu}(q^{-1})u_n = G_{yu}(q^{-1})G_{u\delta}(q^{-1})\delta_n \\ &= \frac{r}{1 - pq^{-1}}\frac{1}{1 - q^{-1}}\delta_n \end{aligned}$$

This can be written as a transfer function form as

$$\frac{y_n}{\delta_n} = G_{y\delta}(q^{-1}) = \frac{r}{(1 - pq^{-1})(1 - q^{-1})}$$

Therefore, the step response of y_n via $G_{yu}(q^{-1})$ can be transformed to impulse response of $G_{y\delta}(q^{-1})$. We can then use one of the three methods as introduced in the previous lectures to find the impulse response of $G_{y\delta}(q^{-1})$ which represents the step response of $G_{yu}(q^{-1})$.

STABILITY ANALYSIS

As discussed in the previous lecture, for a first order discrete transfer function

$$G(q^{-1}) = \frac{r}{1 - pq^{-1}}$$

the impulse response can be written as $y_n = rp^n$. Therefore, if $|p| > 1$, the impulse response, y_t, is unbounded or unstable. We notice that p is the root of the denominator of $G(q^{-1})$ in terms of q. In other words, if we solve $1 - pq^{-1} = 0$, then $q = p$ is the root. The roots of the denominator of a transfer function are known as poles, and the roots of the numerator are known as zeros. Therefore, for a first order discrete transfer function to be stable, the poles should be less than 1. For a higher order discrete transfer function, we can always write it as the summation of first order transfer functions by using partial fraction expansion, i.e.

$$\begin{aligned} \frac{y_n}{u_n} &= \frac{V(q^{-1})}{(1 - p_1q^{-1})(1 - p_2q^{-1}) \cdots (1 - p_mq^{-1})} \\ &= \frac{r_1}{1 - p_1q^{-1}} + \frac{r_2}{1 - p_2q^{-2}} + \cdots + \frac{r_m}{1 - p_mq^{-1}} \end{aligned}$$

Therefore the stability of the high order discrete transfer function requires stability of all the first order transfer functions. The pole of each first order transfer function is one of poles of the high order discrete transfer function. The values of these poles can be complex. Therefore, the stability of the high order discrete transfer function requires all poles within the unit circle (i.e. $|p_i| < 1$ for all i). For example, $q = 2j$ is a complex pole outside unit circle and is therefore an unstable pole. Consider a pole $q = 0.9 + j0.8$. Since $|q| = \sqrt{0.9^2 + 0.8^2} > 1$, this is an unstable pole.

MODIFIED ROUTH STABILITY CRITERION

(Seborg et al., 1989)

In ChE446 or equivalent course, the Routh stability test was used to analyze the denominator of the continuous-time transfer function for unstable roots that lie in the right half of the complex s plane. The denominator of the continuous-time transfer function is also regarded as the characteristic equation. For sampled data systems, this test can also be applied to determine whether any poles of the discrete transfer function (in q) or roots of the characteristic equation lie outside the unit circle. If a simple rational transformation from q to s can be found that maps the interior of the unit circle into the left half of the complex plane, then the Routh criterion can be used directly with discrete transfer functions. Such a transformation (or mapping) is provided by the bilinear transformation

$$q = \frac{1 + w}{1 - w}$$

or

$$w = \frac{q - 1}{q + 1}$$

It can be shown that if $|q| > 1$, then $w > 0$ and if $|q| < 1$ then $|w| < 0$.

To apply this stability test, first determine the denominator (arranged according to descending order of q) as

$$\Gamma(q) = a_m q^m + a_{m-1} q^{m-1} + \cdots + a_1 q + a_0$$

Using the bilinear transformation, the characteristic equation is transformed to a function of w:

$$\Gamma(w) = \bar{a}_m w^n + \bar{a}_{m-1} w^{m-1} + \cdots + \bar{a}_1 w + \bar{a}_0 \qquad (1)$$

also yielding a polynomial in w, where $\bar{a}_i$= real constant coefficient $(i = 0, 1, \ldots, m)$. Note that the $\bar{a}_i$ are not necessarily equal to the a_i. The Routh test can then be applied directly to equation (1) to determine the number of roots of $\Gamma(q)$ lie outside the unit circle.

STEADY STATE VALUE

Given a discrete transfer function:

$$\frac{y_n}{u_n} = \frac{b_1 q^{-1} + \cdots + b_r q^{-r}}{1 + a_1 q^{-1} + \cdots + a_m q^{-m}} \qquad (2)$$

Write it as the difference equation form as

$$y_n + a_1 y_{n-1} + \cdots + a_n y_{n-m} = b_1 u_{n-1} + \cdots + b_r u_{n-r}$$

The steady state of the difference equation is achieved when
$u_n = u_{n-1} = \cdots = u_{n-r} = \bar{u}$ and
$y_n = y_{n-1} = \cdots = y_{n-m} = \bar{y}$, i.e.

$$\bar{y} + a_1 \bar{y} + \cdots + a_m \bar{y} = b_1 \bar{u} + \cdots + \bar{b}_r u$$

Therefore, the steady state value of $\bar{y}$ can be calculated as

$$\bar{y} = \frac{b_1 + \cdots + b_r}{1 + a_1 + \cdots + a_m} \bar{u}$$

Compared to equation (2), we can see that the steady state can be solved by setting $q = 1$ in the discrete transfer function (2).

FINAL VALUE AND INITIAL VALUE THEOREMS

The final or large-time value of a discrete-time response can be found from its transfer function:

$$\lim_{n\to\infty} y_n = \lim_{q^{-1}\to 1} (1-q^{-1}) G_{y\delta}(q^{-1})$$

where δ is a unit impulse signal. For example, if a transfer function is $y_n = G_{yu}(q^{-1})u_n$ and the input u_n is generated from an impulse via $u_n = G_{u\delta}(q^{-1})\delta_n$, then the final value will be

$$\lim_{n\to\infty} y_n = \lim_{q^{-1}\to 1} (1-q^{-1}) G_{yu}(q^{-1}) G_{u\delta}(q^{-1})$$

If the input u_n is a unit-step signal, i.e.

$$u_n = \frac{1}{1-q^{-1}}\delta_n$$

then the final value of the step response is

$$\begin{aligned}\lim_{n\to\infty} y_n &= \lim_{q^{-1}\to 1} (1-q^{-1}) G_{yu}(q^{-1}) \frac{1}{1-q^{-1}} \\ &= \lim_{q^{-1}\to 1} G_{yu}(q^{-1})\end{aligned}$$

which is the steady state gain as discussed in the previous lectures.

Similarly, the initial value of a response is defined as the first step of the output, i.e. y_0, if the signal u_n is applied from the time zero. The initial value of a discrete-time response can be calculated from

$$\lim_{n\to 0} y_n = \lim_{q^{-1}\to 0} G_{y\delta}$$

If there are time-delays in the transfer function, then the initial value will be always zero. For example, for a first order transfer function with time-delays

$$G_{yu} = \frac{q^{-1}}{1 - 0.5q^{-1}}$$

the initial value is zero.

PHYSICAL REALIZABILITY

Physical realizability implies that a discrete-time system cannot have an output signal that depends upon future inputs. If we can write a discrete transfer function (in the backshift operator form) as:

$$G_{yu} = \frac{b_0 + b_1 q^{-1} + \cdots + b_r q^{-r}}{1 + a_1 q^{-1} + \cdots + a_m q^{-m}}$$

i.e., if the leading term in the denominator is normalized as 1, then no terms with positive power of q appear in the numerator. This system is then physical realizable (why?). On the other hand, if a discrete transfer function can only be written as

$$G_{yu} = \frac{q^g (b_0 + b_1 q^{-1} + \cdots + b_r q^{-r})}{1 + a_1 q^{-1} + \cdots + a_m q^{-m}}$$

where $g > 0$ is a positive integer, i.e., at least a term with positive power of q appears in the numerator, then the system is physically unrealizable (why?).

Alternatively, if we write a discrete transfer function in the infinite series by long division or partial fraction expansion, then the physical realizability requires that the first term of

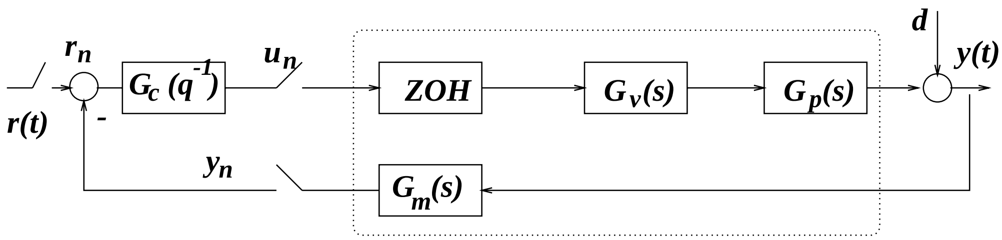


Figure 1: Block diagram of sampled-data control system

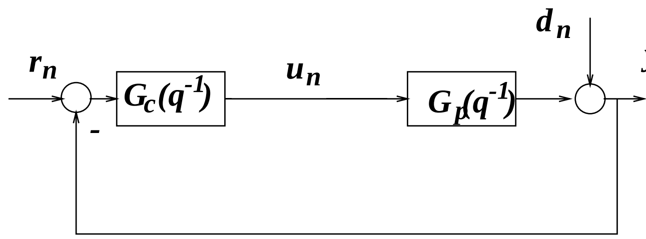


Figure 2: Block diagram of discrete control system

this infinite series begins from a term with zero or negative power of q. For example,

$$G_{yu} = 0.75 + 0.5q^{-1} + 0.25q^{-2} + \cdots +$$

is physically realizable, but

$$G_{yu} = q + 0.75 + 0.5q^{-1} + \cdots +$$

is physically unrealizable.

DEVELOPMENT OF CLOSED-LOOP TRANSFER FUNCTIONS

Consider a closed-loop system shown in Figure 1, where $G_p(s)$ represents the continuous plant transfer function, $G_v(s)$ represents the continuous transfer function of the final control element (e.g., valve), $G_m(s)$ represents the continuous transfer function of the measurement device, and $G_c(q^{-1})$ represents the discrete controller transfer function. We cannot directly write down the closed-loop transfer function from this block diagram since some blocks have continuous transfer functions and the controller block has a discrete transfer function. In order to connect all these blocks, we have to transfer all the continuous transfer functions into discrete transfer functions. As discussed in the previous lectures, we can transfer a continuous transfer function to the discrete transfer function using either finite difference method or exact discretization method. The exact discretization method is preferred but valid only when a zero-order hold exists in the D/A device. Since most control systems use zero-order hold in the D/A device, the exact discretization method is most often used to transfer a continuous transfer function into the discrete transfer function.

In Figure 1, $r(t)$ and $y(t)$ are continuous signals, and r_n and y_n are discrete signals. The closed-loop discrete transfer function will reflect the relationship between r_n and y_n (not between $r(t)$ and $y(t)$). Similarly, in the discrete representation of the plant, the plant input should be a discrete signal that is u_n, and the plant output should also be a discrete signal that is y_n. Therefore, the discrete plant transfer function is the discrete representation of $G_v(s)G_p(s)G_m(s)$ with the zero-order hold. Using the exact discretization method (Matlab function *c2d*), one can transfer the continuous plant transfer function $G_v(s)G_p(s)G_m(s)$ to a single discrete transfer function $G_p(q^{-1})$. The discrete block diagram can therefore be represented by Figure 2. Following a similar procedure as in the continuous system, we can write the discrete closed-loop transfer function from the setpoint r_n

to the output y_n as

$$\frac{y_n}{r_n} = \frac{G_c(q^{-1})G_p(q^{-1})}{1 + G_c(q^{-1})G_p(q^{-1})}$$

The stability of the closed-loop system depends on the poles of the closed-loop transfer function. We can use the modified Routh stability criterion to test the stability of the closed-loop system.

The discrete closed-loop transfer function from the disturbance to the output is more complicated. A simplified discrete representation is often used as shown in Figure 2, where d_n is the fictitiously sampled data of $d(t)$. Therefore, the closed-loop transfer function from a_n to y_n can be written as

$$\frac{y_n}{d_n} = \frac{1}{1 + G_c(q^{-1})G_p(q^{-1})}$$

Notice that the stability conditions are the same for both closed-loop transfer functions.