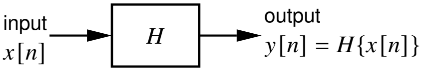
5. Discrete-time systems

- DT systems
- classifications of DT systems
- recursive solution of difference equations
- continuous to discrete signal processing

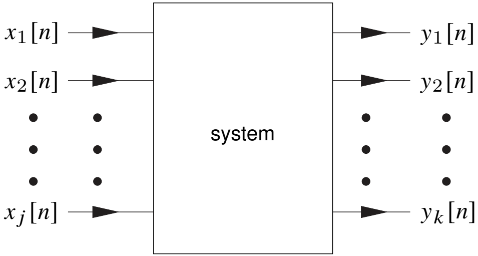
Discrete-time system

a *discrete-time system* is a system whose inputs and outputs are DT signals

Single-input single-output



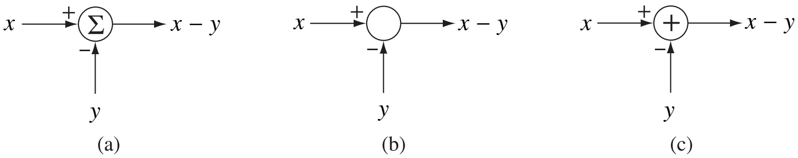
Multi-input multi-output



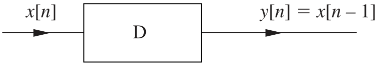
Block diagrams operations

Amplifier (scalar multiplication)

Summation (addition)



Delay



Example 5.1 (saving accounts)

- a person makes a deposit in a bank regularly every $T = 1$ month
- the bank pays a interest rate r on the account balance during the period T

find the equation relating the output $y[n]$ (the balance) to the input $x[n]$ (the deposit)

Solution:

- current balance $y[n]$ is the sum of previous balance $y[n-1]$, the interest on $y[n-1]$ during the period T, and the deposit $x[n]$:

$$\begin{aligned} y[n] &= y[n-1] + ry[n-1] + x[n] \\ &= (1+r)y[n-1] + x[n] \end{aligned}$$

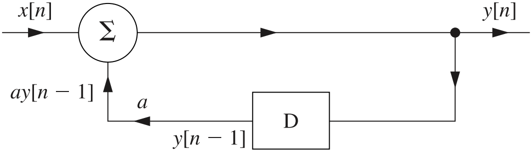
or

$$y[n] - ay[n-1] = x[n], \quad a = 1 + r$$

the equation

$$y[n] - ay[n-1] = x[n]$$

can be represented in block diagram as:

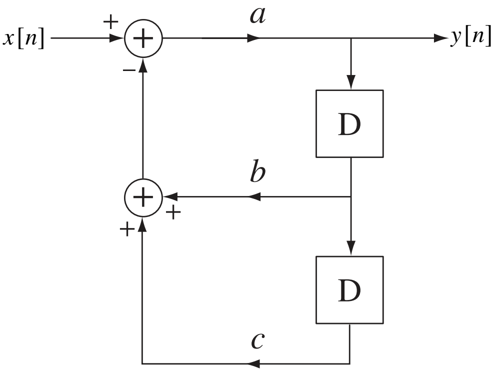


note that so we can replace n by $n+1$ to obtain the advance-form:

$$y[n+1] - ay[n] = x[n+1]$$

Example 5.2

find the input-output relation of the system described shown below



Solution:

$$y[n] = a(x[n] - by[n-1] - cy[n-2])$$

Example 5.3 (sales estimate)

- $x[n]$ students enroll in a course requiring a textbook during semester n
- $y[n]$ new copies of the book are sold by publisher during semester n
- one-quarter of students resell the texts at end of semester
- the book life is three semesters

find the equation relating the new books sold , $y[n]$, to the number of students enrolled in the nth semester, $x[n]$, assuming that every student buys a book

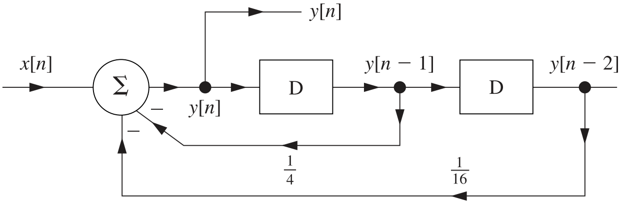
Solution: in the nth semester,

- no. of sold books $y[n]$ and old resold books must be equal to no. of students $x[n]$
- $(1/4)y[n-1]$ books will be resold in the nth semester
- $(1/4)y[n-2]$ books will be resold in semester $(n-1)$ and a quarter of these $(1/16)y[n-2]$ will be resold in the nth semester

hence, $x[n]$ is equal to:

$$y[n] + \tfrac{1}{4}y[n-1] + \tfrac{1}{16}y[n-2] = x[n]$$

the above equation is in delay-form; the block-diagram representation is



in advance form, we can replace n by $n + 2$ to obtain

$$y[n+2] + \tfrac{1}{4}y[n+1] + \tfrac{1}{16}y[n] = x[n+1]$$

Outline

- DT systems
- **classifications of DT systems**
- recursive solution of difference equations
- continuous to discrete signal processing

Linear systems

a DT system is

- *homogeneous* if $x \to y$, then $\alpha x \to \alpha y$ for any number a
- *additive* if for $x_1 \to y_1$ and $x_2 \to y_2$, we have $x_1 + x_2 \to y_1 + y_2$

a system is **linear** if it satisfies the *superposition* property:

$$x_1 \longrightarrow y_1, \quad x_2 \longrightarrow y_2$$

then for any numbers α_1, α_2:

$$\alpha_1 x_1 + \alpha_2 x_2 \longrightarrow \alpha_1 y_1 + \alpha_2 y_2$$

in other words, the system is both homogeneous and additive

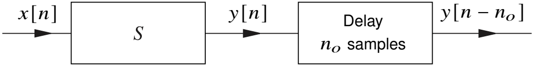
(a system is *nonlinear* if it doesn't satisfy either the homogeneity or additivity property)

Time-invariant systems

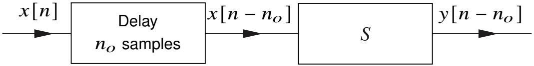
a system is *time-invariant* (*shift invariance*) if

$$x[n] \to y[n] \quad \Longrightarrow \quad x[n-n_o] \to y[n-n_o]$$

for any integer n_o (assuming initial conditions are also delayed by n_o)



(a)



(b)

- a system is *time-varying* if the the above does not hold
- DT system that is both linear and time-invariant is called *linear time-invariant discrete system* (LTID)

Example 5.4

determine whether the system described by $y[n] = e^{-n}x[n]$ is

(a) linear or non-linear

(b) time-invariant or time-varying

Solution:

(a) we have $x_1[n] \longrightarrow y_1[n] = e^{-n}x_1[n]$ and $x_2[n] \longrightarrow y_2[n] = e^{-n}x_2[n]$; for input $x[n] = \alpha_1 x_1[n] + \alpha_2 x_2[n]$, the output is

$$e^{-n}[\alpha_1 x_1[n] + \alpha_2 x_2[n]] = \alpha_1 e^{-n}x_1[n] + \alpha_2 e^{-n}x_2[n] = \alpha_1 y_1[n] + \alpha_2 y_2[n]$$

hence the system satisfies the superposition property, hence linear

(b) we have $x_1[n] \longrightarrow y_1[n] = e^{-n}x[n]$ and input $x_2[n] = x_1[n-n_0]$ gives output

$$y_2[n] = e^{-n}x_2[n] = e^{-n}x_1[n-n_0] \neq y_1[n-n_0]$$

hence, the system is time-varying

Causal and static systems

Causal system: a system is *causal* (or *physical* or *nonanticipative*) if output at $n = k$ depends only on the input $x[n]$ for $n \leq k$

- output depends only on the past and present values of the input
- output does not depends on future inputs
- a system that violates this condition is called a *noncausal* (or *anticipative*) system

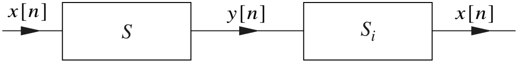
Static (memoryless, instantaneous) system: a system is *static* if the output at any instant n depends on input at n only; otherwise, the system is *dynamic* or *with memory*

Examples

- $y[n] = \sin(x[n])$ is a causal and static (memoryless) system
- $y[n] = x[n] + y[n-1]$ is a causal and dynamic system (with memory)

Invertible systems

a DT system is *invertible* if we can recover input $x[n]$ from output $y[n]$ by another system (called the *inverse system*)



- for an invertible system, every input have a unique output
- example: $y[n] = x[n + k]$ is invertible since $x[n] = y[n - k]$

Noninvertible system

- a system is *noninvertible* when we cannot obtain input from output
- example: $y[n] = x[Mn]$ loses all but every Mth sample of input, hence, noninvertible
- systems where two different inputs give same output are noninverible
  - $y[n] = \cos(x[n])$
  - $y[n] = |x[n]|$

BIBO stable

a system is *bounded-input-bounded-output (BIBO) stable* (externally stable) if every bounded input applied at the input terminal results in a bounded output

Examples:

- $y[n] = x[-n]$ is BIBO stable
- $y[n] = e^{-n}x[n]u[n]$ is BIBO stable
- $y[n] = nx[n]$ is BIBO unstable
- $y[n] = e^{-n}x[n]$ is BIBO unstable

Example 5.5

consider a DT system $y[n] = x[n]x[n-1]$; determine whether the system is

(a) linear

(b) time-invariant

(c) causal

(d) memoryless (static)

(e) invertible

(f) BIBO-stable

Solution: suppose $x[n] \to y[n]$

(a) we have

$$ax[n] \to a^2x[n]x[n-1] = a^2y[n] \neq ay[n]$$

hence, system is nonlinear since it does not satisfy the homogeneity property

(b) we have

$$x[n-n_0] \to x[n-n_0]x[n-1-n_0] = x[n-n_0]x[(n-n_0)-1] = y[n-n_0]$$

hence, the system is time-invariant

(c) causal because the current output does not depend on future input values

(d) output $y[n]$ depends on the past values of input $x[n-1]$; hence not memoryless

(e) two different inputs $x_1[n] = 1$ and $x_2[n] = -1$ give same output $y_1[n] = y_2[n] = 1$, hence noninvertible

(f) for $|x[n]| \leq M_x < \infty$, we have $|y[n]| \leq M_x^2 < \infty$; hence, system is BIBO-stable

Outline

- DT systems
- classifications of DT systems
- **recursive solution of difference equations**
- continuous to discrete signal processing

Linear difference equation

Advance-form

$$y[n+N] + a_1 y[n+N-1] + \cdots + a_{N-1} y[n+1] + a_N y[n] =$$
$$b_0 x[n+M] + b_1 x[n+M-1] + \cdots + b_{M-1} x[n+1] + b_M x[n]$$

- order is N
- system is linear, and time-invariant if a_i, b_i are constants (independent of n)
- system is causal if $M \leq N$
- many systems can be modeled as linear difference systems

Causal delay-form ($M = N$)

$$y[n] + a_1 y[n-1] + \cdots + a_{N-1} y[n-N+1] + a_N y[n-N] =$$
$$b_0 x[n] + b_1 x[n-1] + \cdots + b_{N-1} x[n-N+1] + b_N x[n-N]$$

- delay form is more natural because delay operation is causal, hence realizable
- advance form is more mathematically convenience compared to delay form

Recursive (iterative) solution

we can express the difference equation

$$\begin{aligned} & y[n] + a_1 y[n-1] + \cdots + a_{N-1} y[n-N+1] + a_N y[n-N] \\ & = b_0 x[n] + b_1 x[n-1] + \cdots + b_{N-1} x[n-N+1] + b_N x[n-N] \end{aligned}$$

in *recursive form*:

$$\begin{aligned} y[n] = & - a_1 y[n-1] - a_2 y[n-2] - \cdots - a_N y[n-N] \\ & + b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N] \end{aligned}$$

- to find $y[0]$, we need to know the N initial conditions $y[-1], y[-2], \ldots, y[-N]$
- to find $y[1]$, we need to know the N i.c $y[0], y[-1], \ldots, y[-N+1]$ ...etc
- knowing the N initial conditions and the input, we can determine recursively the entire output $y[0], y[1], y[2], y[3], \ldots$, one value at a time

Example 5.6

solve iteratively (recursively) the equation

$$y[n] - 0.5y[n-1] = x[n]$$

given $y[-1] = 16$ and causal input $x[n] = n^2u[n]$

Solution: the equation can be expressed as

$$y[n] = 0.5y[n-1] + x[n]$$

if we set $n = 0$, we obtain

$$\begin{aligned}
y[0] &= 0.5y[-1] + x[0] = 0.5(16) + 0 = 8 \\
y[1] &= 0.5(8) + (1)^2 = 5 \\
y[2] &= 0.5(5) + (2)^2 = 6.5 \\
y[3] &= 0.5(6.5) + (3)^2 = 12.25 \\
y[4] &= 0.5(12.25) + (4)^2 = 22.125 \\
&\vdots
\end{aligned}$$

Example 5.7

solve iteratively

$$y[n+2] - y[n+1] + 0.24y[n] = x[n+2] - 2x[n+1]$$

with initial conditions $y[-1] = 2$, $y[-2] = 1$ and a causal input $x[n] = nu[n]$

Solution: the system equation can be expressed as

$$y[n+2] = y[n+1] - 0.24y[n] + x[n+2] - 2x[n+1]$$

hence

$$y[0] = y[-1] - 0.24y[-2] + x[0] - 2x[-1] = 2 - 0.24(1) + 0 - 0 = 1.76$$
$$y[1] = y[0] - 0.24y[-1] + x[1] - 2x[0] = 1.76 - 0.24(2) + 1 - 0 = 2.28$$
$$y[2] = y[1] - 0.24y[0] + x[2] - 2x[1] = 2.28 - 0.24(1.76) + 2 - 2(1) = 1.8576$$
$$\vdots$$

Matlab example

- the previews example, can be solved via Matlab code:

```
n = -2:5; y = [1,2,zeros(1,length(n)-2)]; x = [0,0,n(3:end)];
for k = 1:length(n)-2,
y(k+2) = y(k+1)-0.24*y(k)+x(k+2)-2*x(k+1);
end
n,y
```

  output:

```
n = -2      -1      0       1       2       3        4        5
y = 1.0000 2.0000 1.7600 2.2800 1.8576 0.3104 -2.1354 -5.2099
```

- in the example on page 5.4, we can determine the money earned by investing $\$100$ monthly at 0.5% interest per month for 100 months:

```
r = 0.005; a1 = -(1+r); y(1) = 100;
for n = 2:100, y(n) = -a1*y(n-1)+100; end
y(100)-100*100
```

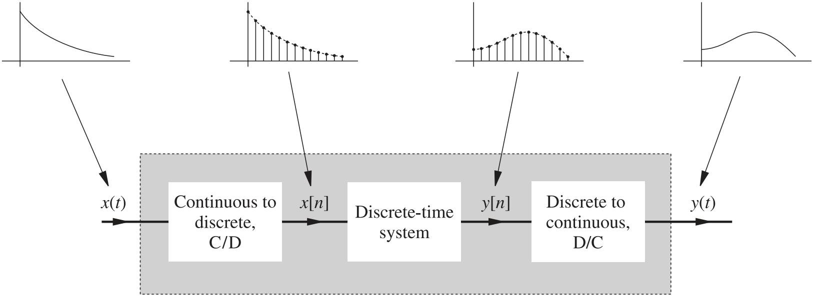
  [output: `ans = 2933.37`]

Outline

- DT systems
- classifications of DT systems
- recursive solution of difference equations
- **continuous to discrete signal processing**

Continuous to discrete C/D and discrete to continuous D/C

- DT system can be used to process a continuous-time signal
- DT signals are more efficient to send, receive, and store information
- DT systems are easier to handle and manipulate compared to CT systems

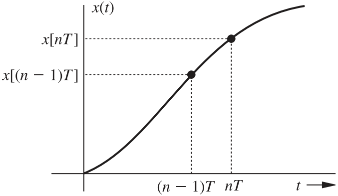


Example: digital differentiator

a *digital differentiator* is a system that differentiates a CT signals by DT processing

- $y(t) = \frac{dx(t)}{dt}$
- let $x[n] = x(nT)$ and $y[n] = y(nT)$

$$y(nT) = \left.\frac{dx(t)}{dt}\right|_{t=nT} = \lim_{T\to 0} \frac{x(nT) - x((n-1)T)}{T}$$

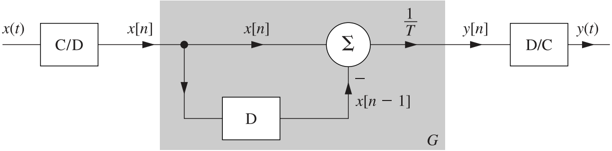


hence

$$y[n] = \lim_{T\to 0} \frac{1}{T}(x[n] - x[n-1])$$

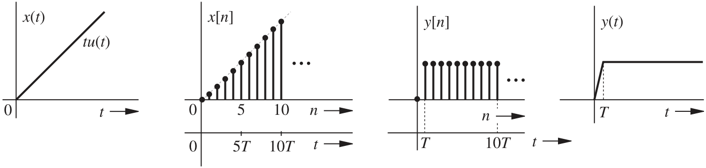
for $T > 0$ sufficiently small, we approximate the differentiator by:

$$y[n] \approx \frac{1}{T}(x[n] - x[n-1])$$



Example: on ramp function $x[n] = nTu[n]$, we get

$$y[n] = \frac{1}{T}(x[n] - x[n-1]) = u[n-1]$$



as T approaches zero, $y(t)$ approaches the desired output $u(t)$

Example: differential equation to difference equation

$$\frac{dy(t)}{dt} + cy(t) = x(t)$$

from the definition of a derivative, we can express the above at $t = nT$ as

$$\lim_{T \to 0} \frac{y[n] - y[n-1]}{T} + cy[n] = x[n]$$

assuming very small $T \neq 0$, we can approximate the above as:

$$y[n] + \alpha y[n-1] = \beta x[n]$$

where

$$\alpha = \frac{-1}{1 + cT}, \qquad \beta = \frac{T}{1 + cT}$$

(a computer solves differential equations by solving an equivalent difference equation; hence, it is important to know how to solve difference systems)

Example: estimation position from video

- a car is filmed using a camera operating at 60 frames per second
- let n designate the film frame, where $n = 0$ corresponds to engine ignition
- by analyzing each frame of the film, we can determine the car position $x[n]$, measured in meters, from the original starting position $x[0] = 0$
- the velocity is $v(t) = \frac{d}{dt}x(t)$ and the acceleration is $a(t) = \frac{d}{dt}v(t)$
- we can estimate the car velocity from the film data by using the difference equation

$$v[n] = 60(x[n] - x[n-1])$$

- we can now estimate the car acceleration from the film data by using

$$a[n] = 60(v[n] - x[v-1])$$

- combining last two equations, we get

$$a[n] = 60(60(x[n] - x[n-1]) - 60(x[n-1] - x(n-2]))$$

  or

$$a[n] = 3600(x[n] - 2x[n-1] + x[n-2])$$

- this estimate of acceleration has two primary advantages;
  - it is simple to calculate
  - it is a causal, stable, LTI system (easy to analyze)

Second order derivative

for small enough T, we can approximate

$$
\begin{aligned}
\left.\frac{dy(t)}{dt}\right|_{t=nT} &\approx \frac{y[n]-y[n-1]}{T} \\
\left.\frac{d^2y(t)}{dt^2}\right|_{t=nT} &= \lim_{T\to 0}\frac{1}{T}\left(\left.\frac{d}{dt}y(t)\right|_{t=nT} - \left.\frac{d}{dt}y(t)\right|_{t=(n-1)T}\right) \\
&\approx \frac{1}{T}\left(\frac{y[n]-y[n-1]}{T} - \frac{y[n-1]-y[n-2]}{T}\right) \\
&= \frac{1}{T^2}\left(y[n]-2y[n-1]+y[n-2]\right)
\end{aligned}
$$

Example 5.8

assuming a sampling interval $T = 0.1$, determine a difference equation model for

$$\ddot{y}(t) + 4\dot{y}(t) + 3y(t) = 100x(t)$$

with initial conditions $y(0) = 0$ and $\dot{y}(0) = 10$

Solution: the differential equation is approximated as

$$\frac{1}{T^2}(y[n] - 2y[n-1] + y[n-2]) + \frac{4}{T}(y[n] - y[n-1]) + 3y[n] = 100x[n]$$

combining terms and substituting $T = 0.1$ yield

$$143y[n] - 240y[n-1] + 100y[n-2] = 100x[n]$$

to compute the equivalent initial conditions, we note that $y[0] = y(0) = 0$; further,

$$10 = \dot{y}(0) = \left.\frac{d}{dt}y(t)\right|_{t=0} \approx \frac{y(0) - y(0-T)}{T} = \frac{y[0] - y[-1]}{0.1} = -10y[-1]$$

Example: digital integrator

we can approximate $y(t) = \int_{-\infty}^{t} x(\tau)d\tau$ using discrete-time systems; at $t = nT$,

$$y(nT) = \lim_{T\to 0} \sum_{k=-\infty}^{n} x(kT)T \quad \text{or} \quad y[n] = \lim_{T\to 0} T \sum_{k=-\infty}^{n} x[k]$$

assuming T is small enough, we get the approximation

$$y[n] = T \sum_{k=-\infty}^{n} x[k]$$

- the above is an example of *accumulator system*
- digital integrator equation can be expressed in the *recursive form*:

$$y[n] - y[n-1] = Tx[n]$$

Digital signal processing (DSP)

advantages of DSP

- digital systems are less sensitive to changes in signal values, thus less sensitive changes in the component parameter values due to temperature variation, aging, and other factors
- digital systems are extremely flexible and easy to implement; digital filter function is easily altered by simply changing the program
- even in the presence of noise, reproduction with digital messages is extremely reliable, often without any deterioration; further, digital signals can be coded to yield extremely low error rates, high fidelity, error correction capabilities, and privacy
- digital signals can be coded to yield extremely low error rates and high fidelity, as well as privacy; also, more sophisticated signal-processing algorithms can be used to process digital signals

- digital filters can be easily time-shared and therefore can serve a number of inputs simultaneously; moreover, it is easier and more efficient to multiplex several digital signals on the same channel
- reproduction with digital messages is extremely reliable without deterioration; analog messages such as photocopies and films, for example, lose quality at each successive stage of reproduction and have to be transported physically from one distant place to another, often at relatively high cost

disadvantages of DSP

- increased system complexity due to use of A/D and D/A interfaces,
- limited range of frequencies available in practice (affordable rates are gigahertz or less)
- use of more power than is needed for the passive analog circuits

References

- B.P. Lathi, *Linear Systems and Signals*, Oxford University Press.
- M. J. Roberts, *Signals and Systems: Analysis Using Transform Methods and MATLAB*, McGraw Hill.