Chapter 2:
Time-domain Models of Systems

In this Chapter we look at systems in input-output form and find ways of ways of solving the system equations to give analytic expressions for the outputs. The main methods we will use are

- Recursive methods (for discrete-time systems only)
- Convolution (for both discrete-time and continuous-time systems)
- Numerical solution methods (for continuous-time systems only)

Definition (N-th order difference equation)

Let N be a positive integer. We define the **N-th order causal linear time-invariant input/output difference equation** to be

$$y[n] + \sum_{i=1}^{N} a_i y[n-i] = \sum_{i=0}^{N} b_i x[n-i], \quad \text{for } n \geq 0$$

where

- $x[n]$ and $y[n]$ are the inputs and outputs, respectively;
- the coefficients a_i and b_i are assumed to be constants.

Algorithm (Recursive solution)

Given N initial conditions $y[-1],\ y[-2],\ \ldots, y[-N]$, and for a given input $x[n]$ with $N+1$ initial conditions $x[0], x[-1],\ x[-2],\ \ldots, x[-N]$, the N-th order difference equation may be solved by

1. *Set $n = 0$ and obtain $y[0]$ by solving*

$$y[0] = -\sum_{i=1}^{N} a_i y[-i] + \sum_{i=0}^{N} b_i x[-i]$$

2. *Repeat Step 1 for $n = 2, \ldots, N-1$.*
3. *Set $n = N$ and obtain $y[N]$ by solving*

$$y[N] = -\sum_{i=1}^{N} a_i y[N-i] + \sum_{i=0}^{N} b_i x[N-i]$$

This process is called ***N-th order recursion****.*

Example (Bank Account)

The difference equation for the balance of a bank account is:

$$y[n] - (1+i)y[n-1] = x[n], \quad \text{for } n \geq 1$$

Here i is the monthly interest rate, $y[0]$ is the initial balance of the account and $x[n]$ is the monthly deposit, commencing at month $n = 1$ (and hence $x[0] = 0$). Applying the recursive method yields

$$\begin{aligned}
y[1] &= (1+i)y[0] + x[1], \\
y[2] &= (1+i)y[1] + x[2], \\
&= (1+i)^2 y[0] + (1+i)x[1] + x[2] \\
y[3] &= (1+i)y[2] + x[3] \\
&= (1+i)^3 y[0] + (1+i)^2 x[1] + (1+i)x[2] + x[3]
\end{aligned}$$

Thus for $n \geq 1$,
$$y[n] = (1+i)^n y[0] + \sum_{i=1}^{n} (1+i)^{n-i} x[i]$$

Remark

The recursive method often leads to cumbersome expressions for the solution, involving complicated series expressions. Next we see how difference equations can be solved with convolution. In Chapter 7, we will see how to solve them using the method of z-transforms.

Definition (Unit pulse response)

For a discrete-time system defined by the difference equation

$$y[n] + \sum_{i=1}^{N} a_i y[n-i] = \sum_{i=0}^{N} b_i x[n-i], \quad \text{for } n \geq 0$$

the system's **unit pulse response**, denoted by $h[n]$, is defined to be the output of the system when the input is $x[n] = \delta[n]$, and the initial conditions are assumed to be zero: $0 = y[-1] = y[-2] = \cdots = y[-N]$.

Introducing Convolution

Convolution comes with a Health Warning:

Definition (from the Oxford online dictionary)

1. The process of becoming coiled or twisted.
2. A thing that is complex and difficult to follow.
3. A sinuous fold in the surface of the brain.

Definition (Convolution for signals)

Let x and y be discrete-time signals. The we define the **convolution of x and y** as the discrete-time signal

$$(x \star y)[n] = \sum_{i=-\infty}^{\infty} x[i]y[n-i]$$

Definition

For any discrete-time signal y and any integer i, we define the **Delay signal of** y as

$$Delay_i(y[n]) = y[n-i]$$

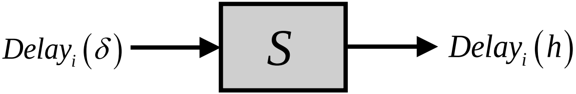
Recall that by the Sifting Property, every discrete-time signal can be represented as

$$x[n] = \sum_{i=-\infty}^{\infty} x[i]\delta[n-i] = \sum_{i=-\infty}^{\infty} x[i] Delay_i(\delta[n])$$

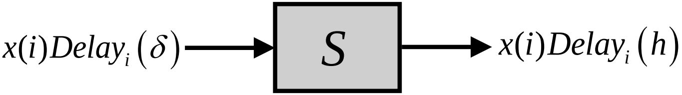
We now use these properties to express the output of a discrete-time system in terms of the pulse response.

- For any linear time-invariant system S, the unit pulse response h is the response to δ:

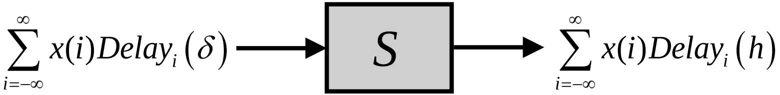
- Since S is time-invariant, the delayed input $Delay_i(\delta)$ produces output $Delay_i(h)$:



- Since S is linear, multiplying an input signal by a real value $x[i]$ will multiply the corresponding output signal by the same value:



- Since S is linear, sums of inputs produce outputs that are sums of those inputs:



- Notice that

$$\sum_{i=-\infty}^{\infty} x[i] Delay_i(h[n]) = \sum_{i=-\infty}^{\infty} x[i]h[n-i] = (x \star h)[n]$$

Theorem

For any LTI discrete-time system, the output y from any input x can be expressed in terms of the convolution of x with the unit pulse response h:

$$y[n] = (x \star h)[n] = \sum_{i=-\infty}^{\infty} x[i]h[n-i]$$

Theorem (Properties of convolution)

- *Commutativity:* $x \star y = y \star x$
- *Associativity:* $(x \star y) \star z = x \star (y \star z)$
- *Distributivity:* $x \star (y + z) = x \star y + x \star z$
- *Homogeneity: For all* $a \in \mathbf{R}$, $\quad x \star (ay) = a(x \star y)$
- *Time-invariance: For all* $N \in \mathbf{Z}$, $Delay_N(x \star y) = x \star (Delay_N(y))$

Theorem (Convolution for finite-duration signals)

Let x and y be finite duration discrete-time signals, such that the support of x is $[n_x, N_x]$, and the support of y is $[n_y, N_y]$. Then their convolution has support $[n_x + n_y, N_x + N_y]$, and is given by

$$(x \star y)[n] = \begin{cases} 0, & \text{if } n < n_x + n_y \\ \sum_{i=n_x}^{N_x} x[i]y[n-i], & \text{if } n_x + n_y \leq n \leq N_x + N_y \\ 0 & \text{if } n > N_x + N_y \end{cases}$$

By the commutativity of convolution, we also have

Corollary

$$(x \star y)[n] = \begin{cases} 0, & \text{if } n < n_x + n_y \\ \sum_{i=n_y}^{N_y} x[n-i]y[i], & \text{if } n_x + n_y \leq n \leq N_x + N_y \\ 0 & \text{if } n > N_x + N_y \end{cases}$$

Definition (The Flip operation)

For any discrete signal $x[n]$, we define the signal $Flip(x)$ as

$$Flip(x[n]) = x[-n]$$

The flip operation reflects the signal in the vertical axis. This can be helpful for computations.

Example (Graphical method for computing a convolution)

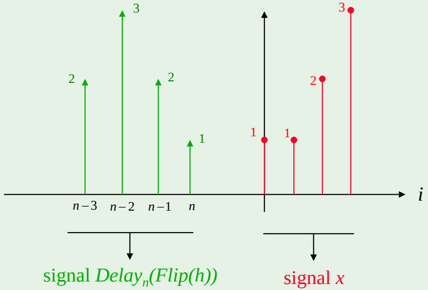
Let x and h be finite-duration signals with

$$\begin{aligned}(x[0],x[1],x[2],x[3]) &= (1,1,2,3) \quad \text{with } x[n]=0 \text{ otherwise}\\ (h[0],h[1],h[2],h[3]) &= (1,2,3,2) \quad \text{with } h[n]=0 \text{ otherwise}\end{aligned}$$

Thus $n_x = n_h = 0$, and $N_x = N_h = 3$. So their convolution will be non-zero only for $0 \le n \le 6$.
We will use the graphical method for computing their convolution.

Example

We plot the signals $Delay_n(Flip(h[i]))$ and $x[i]$ on the same axes:

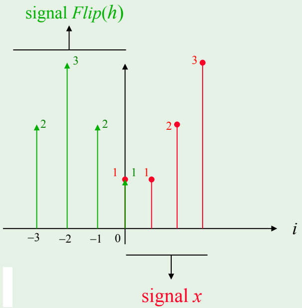


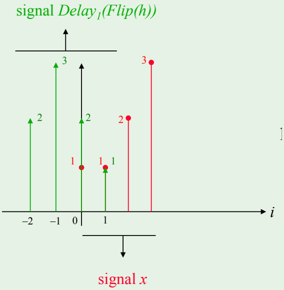
For $n < 0$, there is no overlap in the non-zero values of x and $Delay_n(Flip(h))$, so $x[i]h[n-i] = 0$ for all i. Hence

$$y[n] = \sum_{i=-\infty}^{\infty} x[i]h[n-i] = 0 \quad \text{for } n < 0$$

Example

For $n = 0$, x and $Delay_0(Flip(h))$ overlap at $i = 0$.
For $n = 1$, x and $Delay_1(Flip(h))$ overlap at $i = 0, 1$.





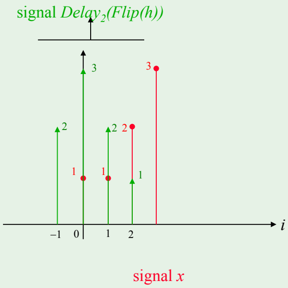
$n = 0$: $n = 1$

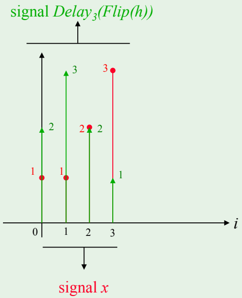
$$y[0] = \sum_{i=-\infty}^{\infty} x[i]h[-i] = x[0]h[0] = (1)(1) = 1$$

$$y[1] = \sum_{i=-\infty}^{\infty} x[i]h[1-i] = x[0]h[1] + x[1]h[0] = (1)(2) + (1)(1) = 3$$

Example

For $n = 2$, x and $Delay_2(Flip(h))$ overlap for $i = 0, 1, 2$.
For $n = 3$, x and $Delay_3(Flip(h))$ overlap for $i = 0, 1, 2, 3$.





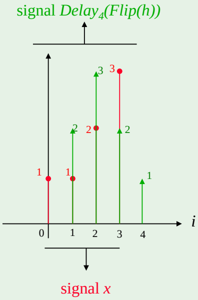
$n = 2$: $n = 3$

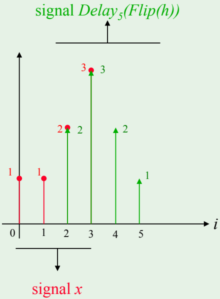
$$y[2] = \sum_{i=-\infty}^{\infty} x[i]h[2-i] = x[0]h[2] + x[1]h[1] + x[2]h[0] = 7$$

$$y[3] = \sum_{i=-\infty}^{\infty} x[i]h[3-i] = x[0]h[3] + x[1]h[2] + x[2]h[1] + x[3]h[0] = 12$$

Example

For $n = 4$, x and $Delay_4(Flip(h))$ overlap for $i = 1, 2, 3$.
For $n = 5$, x and $Delay_5(Flip(h))$ overlap for $i = 2, 3$.





$n = 4$:

$$y[4] = \sum_{i=-\infty}^{\infty} x[i]h[4-i] = x[1]h[3] + x[2]h[2] + x[3]h[1] = 14$$

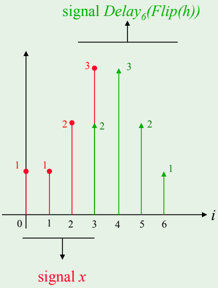
$n = 5$:

$$y[5] = \sum_{i=-\infty}^{\infty} x[i]h[5-i] = x[2]h[3] + x[3]h[2] = 13$$

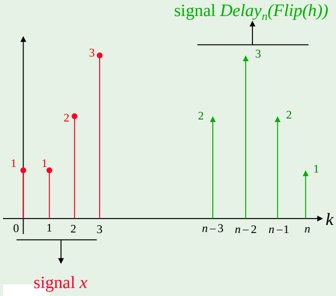
Example

For $n = 6$, x and $Delay_6(Flip(h))$ overlap for $i = 3$.
For $n \geq 7$, x and $Delay_n(Flip(h))$ do not overlap.

$n = 6$:



$n \geq 7$:



$$y[6] = \sum_{i=-\infty}^{\infty} x[i]h[6-i] = x[3]h[3] = 6$$

Another method of computing convolutions is the **Array method.**

Theorem

Let x and y be finite duration discrete-time signals with support $[n_x, N_x]$ and $[n_y, N_y]$ respectively. Then their **convolution array** *is given by*

	$x[n_x]$	$x[n_x+1]$	$\dots$	$x[N_x]$
$y[n_y]$	$x[n_x]y[n_y]$	$x[n_x+1]y[n_y]$	$\dots$	$x[N_x]y[n_y]$
$y[n_y+1]$	$x[n_x]y[n_y+1]$	$x[n_x+1]y[n_y+1]$	$\dots$	$x[N_x]y[n_y+1]$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$y[N_y]$	$x[n_x]y[N_y]$	$x[n_x+1]y[N_y]$	$\dots$	$x[N_x]y[N_y]$

The values of $(x \star y)[n]$ are given by the sums of the elements on the backwards diagonals, where the diagonal beginning at $x[n_x + i]$ and finishing at $y[n_y + i]$ is summed to give $(x \star y)[n_x + n_y + i]$.

Example

Let x and h be the discrete-time signals in the previous example. Their convolution array is

$$
\begin{array}{cc|cccc}
 & & x[0] & x[1] & x[2] & x[3] \\
 & & 1 & 1 & 2 & 3 \\
\hline
h[0] & 1 & 1 & 1 & 2 & 3 \\
h[1] & 2 & 2 & 2 & 4 & 6 \\
h[2] & 3 & 3 & 3 & 6 & 9 \\
h[3] & 2 & 2 & 2 & 4 & 6
\end{array}
$$

Then

$$
\begin{aligned}
y[0] &= 1,\ y[1] = 1+2 = 3,\ y[2] = 2+2+3 = 7; \\
y[3] &= 3+4+3+2 = 12,\ y[4] = 6+6+2 = 14,\ y[5] = 9+4 = 13, \\
y[6] &= 6,\ y[n] = 0, \text{otherwise}
\end{aligned}
$$

Let S be a **causal** linear time-invariant system, with unit pulse response h to the input δ:

Since $\delta[n] = 0$ for $n < 0$, we must have $h[n] = 0$ for $n < 0$. This leads to

Theorem

An LTI system is **causal** *if and only if its unit pulse response h satisfies*

$$h[n] = 0 \quad \textit{for } n < 0$$

For causal systems, we get a simplification for the case that the input signal satisfies $x[n] = 0$ for $n < 0$.

Theorem

Let h be the unit pulse response of a causal LTI system and let $x[n] = 0$ for $n < 0$. Then the output y, for $n \geq 0$, is

$$\begin{aligned}
y[n] = (h \star x)[n] &= \sum_{i=-\infty}^{\infty} x[i]h[n-i] \\
&= \sum_{i=0}^{\infty} x[i]h[n-i], \quad \textit{since } x[i] = 0 \textit{ for } i < 0 \\
&= \sum_{i=0}^{n} x[i]h[n-i], \quad \textit{since } h[n-i] = 0 \textit{ for } i > n
\end{aligned}$$

So the output of a casual system can be expressed as a finite sum.

Definition (N-th order differential equation)

For any positive integer N, we define ***N*-th order causal linear time-invariant input/output differential equation** to be

$$\frac{dy^N}{dt^N} + \sum_{i=0}^{N-1} a_i \frac{dy^i}{dt^i} = \sum_{i=0}^{N} b_i \frac{dx^i}{dt^i} \quad \text{for } t \geq 0$$

where

- $x(t)$ and $y(t)$ are the inputs and outputs, respectively;
- the coefficients a_i and b_i are real constants.

Remark

For differential equations, there is no equivalent solution method to the Recursive method that we used for solving difference equations. Next we see how differential equations can be solved with convolution, and in Chapter 7, we will see how to solve differential equations using the method of Laplace transforms.

Definition (Impulse response)

For a continuous-time system defined by the differential equation

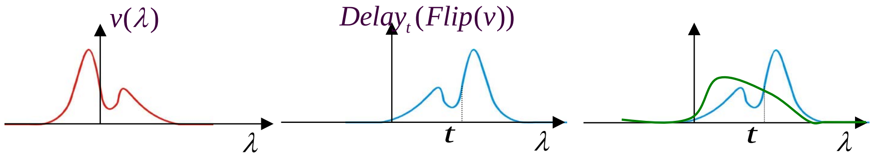
$$\frac{dy^N}{dt^N} + \sum_{i=0}^{N-1} a_i \frac{dy^i}{dt^i} = \sum_{i=0}^{N} b_i \frac{dx^i}{dt^i} \quad \text{for } t \geq 0$$

the system's **impulse response**, denoted by $h(t)$, is defined to be the output of the system when the input is $x(t) = \delta(t)$ (the Dirac delta function), and the initial conditions are assumed to be zero: $0 = y(0) = \dot{y}(0) = \cdots = y^{(N-1)}(0)$.

Definition (Convolution for continuous-time signals)

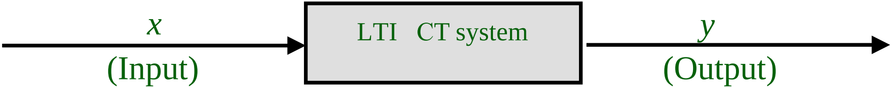
Let x and v be continuous-time signals. The we define the **convolution of** x **and** v as the continuous-time signal

$$(x \star v)(t) = \int_{-\infty}^{\infty} x(\lambda)v(t-\lambda)\, d\lambda$$



Theorem (Properties of continuous-time convolution)

- *Commutativity:* $x \star y = y \star x$
- *Associativity:* $(x \star y) \star z = x \star (y \star z)$
- *Distributivity:* $x \star (y + z) = x \star y + x \star z$
- *Homogeneity: For all* $a \in \mathbf{R}$, $\quad x \star (ay) = a(x \star y)$
- *Time-invariance: For all* $\tau \in \mathbf{R}$, $Delay_\tau(x \star y) = x \star (Delay_\tau(y))$



Theorem

For any LTI continuous-time system, the output y from any input x can be expressed in terms of the convolution of the impulse response h as:

$$y(t) = (x \star h)(t) = \int_{-\infty}^{\infty} h(\lambda)x(t-\lambda)\ d\lambda$$

For causal systems, we have a nice simplification for the case that the input signal satisfies $x(t) = 0$ for $t < 0$.

Theorem

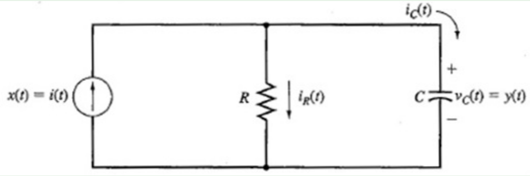
Let h be the impulse response of a causal LTI system and let the input x satisfy $x(t) = 0$ for $t < 0$. Then the output y, for $t \geq 0$ is

$$y(t) = (h \star x)(t) = \int_0^t h(\lambda)x(t-\lambda)\, d\lambda$$

So the output of a casual system can be expressed as a finite integral.

Example (RC circuit)

Recall the *RC* (resistive-capacitive) electric circuit



with input-output differential equation

$$C\frac{dy}{dt} + \frac{1}{R}y(t) = x(t)$$

The impulse response h arises when the input x is the Dirac delta function $\delta(t)$.

Example (RC circuit)

The impulse response can be evaluated as follows:

1. Define, for any $\varepsilon > 0$, the input signal

$$x_\varepsilon(t) = \begin{cases} \frac{1}{\varepsilon}, & \text{for } t \leq \varepsilon \\ 0, & \text{otherwise} \end{cases}$$

2. Solve the differential equation with this input $x_\varepsilon(t)$ to obtain the output $y_\varepsilon(t)$.
3. Obtain $h(t) = \lim_{\varepsilon \to 0} y_\varepsilon(t)$.

See Workshops for the details. If $R = C = 1$, the impulse response for the *RC* circuit is

$$h(t) = \begin{cases} e^{-t}, & \text{for } t \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

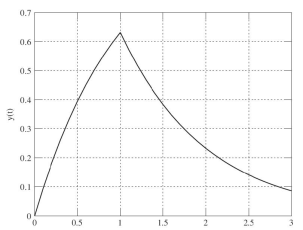
Example (RC circuit)

By convolution, for any input x with $x(t) = 0$ for $t < 0$, the output for $t \geq 0$, is

$$y(t) = (h \star x)(t) = \int_0^t e^{-\lambda} x(t-\lambda)\, d\lambda$$

Suppose the input is the pulse $p(t) = 1$ for $0 \leq t \leq 1$. Then

$$y(t) = (h \star p)(t) = \begin{cases} 1 - e^{-t}, & 0 \leq t \leq 1 \\ e^{-t}(e-1), & t \geq 1 \end{cases}$$



Discrete-time systems are often used as approximations for continuous-time systems. The essential link is provided by Euler:

Definition (Euler approximations)

Let y be a differentiable function and let $T > 0$. For integer n, we define the *Euler approximation* to the first derivative of y as

$$\left.\frac{dy}{dt}\right|_{t=nT} \approx \frac{y(nT+T) - y(nT)}{T}$$

and the Euler approximation to the second derivative of y as

$$\left.\frac{d^2y}{dt^2}\right|_{t=nT} \approx \frac{y(nT+2T) - 2y(nT+T) + y(nT)}{T^2}$$

T is referred to as the *step size*. These approximations become more accurate as we reduce T.

Consider the first-order system defined by the differential equation

$$\frac{dy}{dt} + ay(t) = bx(t) \quad \text{for } t \geq 0$$

For a given step size T, we introduce the discrete signals $y[n] = y(nT)$ and $x[n] = x(nT)$. Using the Euler approximation to $\frac{dy}{dt}$, we can obtain the first order difference equation

$$y[n+1] + (aT - 1)y[n] = bTx[n]$$

This difference equation can then be solved by, for example, the Recursion method to find $y[n]$, the system output y at times $t = nT$. For the system response with initial condition $y(0)$ and zero input $x(t) = 0$, solving the difference equation gives

$$y[n] = (1 - aT)^n y(0)$$

From ODE theory (or using Laplace transforms, see Chapter 7), we know that the exact solution of the differential equation is

$$y(t) = e^{-at}y(0)$$

Thus at time $t = nT$, the exact value of the solution is

$$\begin{aligned} y(nT) &= e^{-anT}y(0) \\ &= (e^{-aT})^n y(0) \\ &= \left(1 - aT + \frac{a^2T^2}{2} - \frac{a^3T^3}{6} + \dots\right)^n y(0) \end{aligned}$$

from Taylor's theorem. Thus the approximation is close when

$$1 - aT \approx e^{-aT}$$

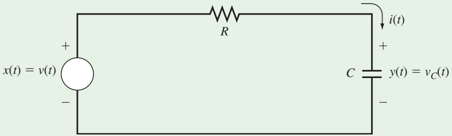
which requires $T^2 \approx 0$.

Example

An RC series circuit can be described by the equation

$$\frac{dy}{dt} + \frac{1}{RC} y(t) = \frac{1}{RC} x(t) \quad \text{for } t \geq 0$$

where x is the input voltage and y is the capacitor voltage.

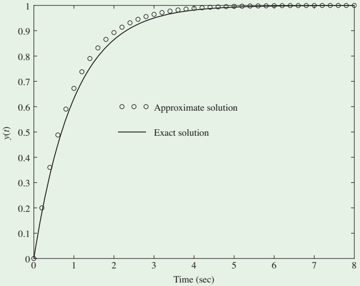


Using Euler approximations, we obtain the difference equation

$$y[n+1] + \left(\frac{T}{RC} - 1\right) y[n] = \frac{T}{RC} x[n]$$

Example

We assume $R = C = 1$, $x(t) = 1$ and $y(0) = 0$ and plot both the exact solution and approximate the solution with $T = 0.2$:



- The approximate solution is close to the exact one, and this can be improved by using smaller T.
- Euler approximations are only one way of discretizing a continuous time system. The *Runge-Kutta method* is widely used and gives closer approximations.