Lecture 1B – Systems

Differential equations. System modelling. Discrete-time signals and systems. Difference equations. Discrete-time block diagrams. Discretization in time of differential equations. Convolution in LTI discrete-time systems. Convolution in LTI continuous-time systems. Graphical description of convolution. Properties of convolution. Numerical convolution.

Linear Differential Equations with Constant Coefficients

Modelling of real systems involves approximating the real system to such a degree that it is tractable to our mathematics. Obviously the more assumptions we make about a system, the simpler the model, and the more easily solved. The more accurate we make the model, the harder it is to analyse. We need to make a trade-off based on some specification or our previous experience.

A lot of the time our modelling ends up describing a continuous-time system that is linear, time-invariant (LTI) and finite dimensional. In these cases, the system is described by the following equation:

Linear differential equation

$$\boxed{y^{(N)}(t)+\sum_{i=0}^{N-1}a_i y^{(i)}(t)=\sum_{i=0}^{M}b_i x^{(i)}(t)} \tag{1B.1}$$

where:

$$y^{(N)}(t)=\frac{d^N y(t)}{dt^N} \tag{1B.2}$$

Initial Conditions

The above equation needs the N initial conditions:

$$\boxed{y(0^-),\, y^{(1)}(0^-),\, \ldots,\, y^{(N-1)}(0^-)} \tag{1B.3}$$

We take 0^- as the time for initial conditions to take into account the possibility of an impulse being applied at $t=0$, which will change the output instantaneously.

First-Order Case

For the first order case we can express the solution to Eq. (1B.1) in a useful (and familiar) form. A first order system is given by:

First-order linear differential equation

$$\boxed{\frac{dy(t)}{dt}+ay(t)=bx(t)} \tag{1B.4}$$

To solve, first multiply both sides by an *integrating factor* equal to e^{at}. This gives:

$$e^{at}\left[\frac{dy(t)}{dt}+ay(t)\right]=e^{at}bx(t) \tag{1B.5}$$

Thus:

$$\frac{d}{dt}\left[e^{at}y(t)\right]=e^{at}bx(t) \tag{1B.6}$$

Integrating both sides gives:

$$e^{at}y(t)-y(0^-)=\int_{0^-}^{t}e^{a\tau}bx(\tau)d\tau,\quad t\geq 0 \tag{1B.7}$$

Finally, dividing both sides by the integrating factor gives:

First glimpse at a convolution integral – as the solution of a first-order linear differential equation

$$y(t)=e^{-at}y(0^-)+\int_{0^-}^{t}e^{-a(t-\tau)}bx(\tau)d\tau,\quad t\geq 0 \tag{1B.8}$$

Use this to solve the simple revision problem for the case of the unit step.

The two parts of the response given in Eq. (1B.8) have the obvious names zero-input response (ZIR) and zero-state response (ZSR). It will be shown later that the ZSR is given by a convolution between the system's impulse response and the input signal.

System Modelling

In modelling a system, we are nearly always after the input/output relationship, which is a differential equation in the case of continuous-time systems. If we're clever, we can break a system down into a connection of simple components, each having a relationship between cause and effect.

Electrical Circuits

The three basic linear, time-invariant relationships for the resistor, capacitor and inductor are respectively:

Cause / effect relationships for electrical systems

$$v(t)=Ri(t) \tag{1B.9a}$$

$$i(t)=C\frac{dv(t)}{dt} \tag{1B.9b}$$

$$v(t)=L\frac{di(t)}{dt} \tag{1B.9c}$$

Mechanical Systems

In linear translational systems, the three basic linear, time-invariant relationships for the inertia force, damping force and spring force are respectively:

Cause / effect relationships for mechanical translational systems

$$F(t)=M\frac{d^2x(t)}{dt} \tag{1B.10a}$$

$$F(t)=k_d\frac{dx(t)}{dt} \tag{1B.10b}$$

$$F(t)=k_s x(t) \tag{1B.10c}$$

Where $x(t)$ is the position of the object under study.

For rotational motion, the relationships for the inertia torque, damping torque and spring torque are:

Cause / effect relationships for mechanical rotational systems

$$F(t) = I\frac{d^2\theta(t)}{dt} \tag{1B.11a}$$

$$F(t) = k_d\frac{d\theta(t)}{dt} \tag{1B.11b}$$

$$F(t) = k_s\theta(t) \tag{1B.11c}$$

Finding an input-output relationship for signals in systems is just a matter of applying the above relationships to a conservation law: for electrical circuits it is one of Kirchhoff's laws, in mechanical systems it is D'Alembert's principle.

Discrete-time Systems

A discrete-time signal is one that takes on values only at discrete instants of time. Discrete-time signals arise naturally in studies of economic systems – amortization (paying off a loan), models of the national income (monthly, quarterly or yearly), models of the inventory cycle in a factory, etc. They arise in science, eg. in studies of population, chemical reactions, the deflection of a weighted beam. They arise all the time in electrical engineering, because of digital control eg. radar tracking system, processing of electrocardiograms, digital communication (CD, mobile phone, Internet). Their importance is probably now reaching that of continuous-time systems in terms of analysis and design – specifically because today signals are processed digitally, and hence they are a special case of discrete-time signals.

Discrete-time systems are important…

It is now cheaper and easier to perform most signal operations inside a microprocessor or microcontroller than it is with an equivalent analog continuous-time system. But since there is a great depth to the analysis and design techniques of continuous-time systems, and since most physical systems are continuous-time in nature, it is still beneficial to study systems in the continuous-time domain.

…especially as microprocessors play a central role in today's signal processing

Linear Difference Equations with Constant Coefficients

Linear, time-invariant, discrete-time systems can be modelled with the difference equation:

$$y[n] + \sum_{i=1}^{N} a_i y[n-i] = \sum_{i=0}^{M} b_i x[n-i] \tag{1B.12}$$

Linear time-invariant (LTI) difference equation

Solution by Recursion

We can solve difference equations by a direct numerical procedure.

There is a MATLAB® function available for download from the Signals and Systems web site called `recur` *that solves the above equation.*

Complete Solution

By solving Eq. (1B.12) recursively it is possible to generate an expression for the complete solution $y[n]$ in terms of the initial conditions and the input $x[n]$.

First-Order Case

Consider the first-order linear difference equation:

$$y[n] + ay[n-1] = bx[n] \tag{1B.13}$$

First-order linear difference equation

with initial condition $y[-1]$. *By successive substitution, show that:*

$$\begin{aligned} y[0] &= -ay[-1] + bx[0] \\ y[1] &= a^2 y[-1] - abx[0] + bx[1] \\ y[2] &= -a^3 y[-1] + a^2 bx[0] - abx[1] + bx[2] \end{aligned} \tag{1B.14}$$

From the pattern, it can be seen that for $n \geq 0$,

First look at a convolution summation – as the solution of a first-order linear difference equation

$$\boxed{y[n] = (-a)^{n+1} y[-1] + \sum_{i=0}^{n} (-a)^{n-i} bx[i]} \tag{1B.15}$$

This solution is the discrete-time counterpart to Eq. (1B.8).

Discrete-Time Block Diagrams

An LTI discrete-time system can be represented as a block diagram consisting of adders, gains and delays. The gain element is shown below:

A discrete-time gain element

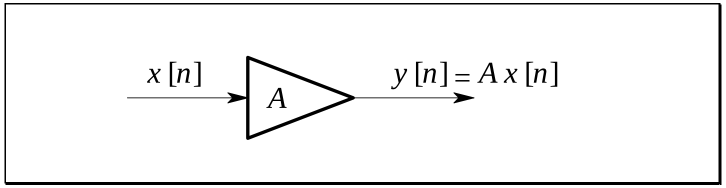


Figure 1B.1

The *unit-delay* element is shown below:

A discrete-time unit-delay element

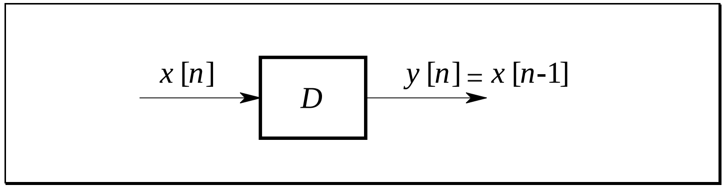
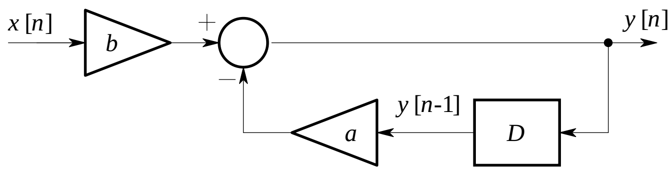


Figure 1B.2

Such an element is normally implemented by the memory of a computer, or a digital delay line.

Example

Using these two elements and an adder, we can construct a representation of the discrete-time system given by $y[n] + ay[n-1] = bx[n]$. The system is shown below:



Discretization in Time of Differential Equations

Often we wish to use a computer for the solution of continuous-time differential equations. We can: if we are careful about interpreting the results.

First-Order Case

Let's see if we can approximate the first-order linear differential equation given by Eq. (1B.4) with a discrete-time equation. We can approximate the continuous-time derivative using *Euler's approximation*, or forward difference:

$$\boxed{\left.\frac{dy(t)}{dt}\right|_{t=nT} \approx \frac{y(nT+T) - y(nT)}{T}} \tag{1B.16}$$

Approximating a derivative with a difference

If T is suitably small and $y(t)$ is continuous, the approximation will be accurate. Substituting this approximation into Eq. (1B.4) results in a discrete-time approximation given by the difference equation:

$$\boxed{y[n] \approx (1 - aT)y[n-1] + bTx[n-1]} \tag{1B.17}$$

The first-order difference equation approximation of a first-order differential equation

The discrete values $y[n]$ are approximations to the solution $y(nT)$.

Show that $y[n]$ gives approximate values of the solution $y(t)$ at the times $t = nT$ with arbitrary initial condition $y[-1]$ for the special case of zero input.

Second-order Case

We can generalize the discretization process to higher-order differential equations. In the second-order case the following approximation can be used:

The second-order difference equation approximation of a second-order derivative

$$\left.\frac{d^2 y(t)}{dt^2}\right|_{t=nT} \approx \frac{y(nT+2T)-2y(nT+T)+y(nT)}{T^2} \tag{1B.18}$$

Now consider the second-order differential equation:

$$\frac{d^2 y(t)}{dt^2}+a_1\frac{dy(t)}{dt}+a_0 y(t)=b_1\frac{dx(t)}{dt}+b_0 x(t) \tag{1B.19}$$

Show that the discrete-time approximation to the solution $y(t)$ *is given by:*

An n^{th}-order differential equation can be approximated with an n^{th}-order difference equation

$$\begin{aligned} y[n] &= \left(2-a_1T\right)y[n-1]-\left(1-a_1T+a_0T^2\right)y[n-2] \\ &\quad +b_1Tx[n-1]+\left(b_0T^2-b_1T\right)x[n-2] \end{aligned} \tag{1B.20}$$

Convolution in Linear Time-invariant Discrete-time Systems

Although the linear difference equation is the most basic description of a linear discrete-time system, we can develop an *equivalent* representation called the convolution representation. This representation will help us to determine important system properties that are not readily apparent from observation of the difference equation.

One advantage of this representation is that the output is written as a linear combination of past and present input signal elements. *It is only valid when the system's initial conditions are all zero.*

First-Order System

We have previously considered the difference equation:

$$y[n]+ay[n-1]=bx[n] \tag{1B.21}$$

Linear difference equation

and showed by successive substitution that:

The complete response

$$y[n]=(-a)^{n+1}y[-1]+\sum_{i=0}^{n}(-a)^{n-i}bx[i] \tag{1B.22}$$

By the definition of the convolution representation, we are after an expression for the output *with all initial conditions zero*. We then have:

The zero-state response (ZSR) – a convolution summation

$$y[n]=\sum_{i=0}^{n}(-a)^{n-i}bx[i] \tag{1B.23}$$

In contrast to Eq. (1B.21), we can see that Eq. (1B.23) depends exclusively on present and past values of the input signal. One advantage of this is that we may directly observe how each past input affects the present output signal. For example, an input $x[i]$ contributes an amount $(-a)^{n-i}bx[i]$ to the totality of the output at the n^{th} period.

Unit-Pulse Response of a First-Order System

A discrete-time system's unit-pulse response defined

The output of a system subjected to a unit-pulse response $\delta[n]$ is denoted $h[n]$ and is called the *unit-pulse response*, or *weighting sequence* of the discrete-time system. It is very important because it completely characterises a system's behaviour. It may also provide an experimental or mathematical means to determine system behaviour.

For the first-order system of Eq. (1B.21), if we let $x[n]=\delta[n]$, then the output of the system to a unit-pulse input can be expressed using Eq. (1B.23) as:

$$y[n]=\sum_{i=0}^{n}(-a)^{n-i}b\delta[i] \tag{1B.24}$$

which reduces to:

$$y[n]=(-a)^{n}b \tag{1B.25}$$

A first-order discrete-time system's unit-pulse response

The unit-pulse response for this system is therefore given by:

$$h[n]=(-a)^{n}bu[n] \tag{1B.26}$$

General System

For a general linear time-invariant (LTI) system, the response to a delayed unit-pulse $\delta[n-i]$ must be $h[n-i]$.

Since $x[n]$ can be written as:

$$x[n]=\sum_{i=0}^{\infty}x[i]\delta[n-i] \tag{1B.27}$$

and since the system is LTI, the response $y_i[n]$ to $x[i]\delta[n-i]$ is given by:

$$y_i[n]=x[i]h[n-i] \tag{1B.28}$$

The response to the sum Eq. (1B.27) must be equal to the sum of the $y_i[n]$ defined by Eq. (1B.28). Thus the response to $x[n]$ is:

$$\boxed{y[n]=\sum_{i=0}^{\infty}x[i]h[n-i]} \tag{1B.29}$$

Convolution summation defined for a discrete-time system

This is the convolution representation of a discrete-time system, also written as:

$$\boxed{y[n]=h[n]*x[n]} \tag{1B.30}$$

Convolution notation for a discrete-time system

Graphically, we can now represent the system as:

Graphical notation for a discrete-time system using the unit-pulse response

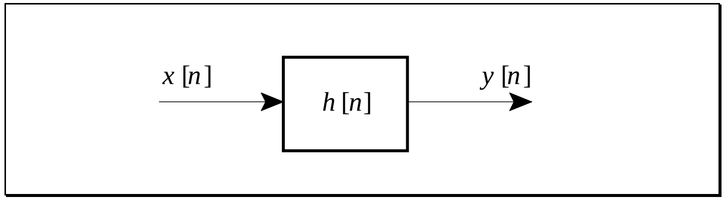


Figure 1B.3

It should be pointed out that the convolution representation is not very efficient in terms of a digital implementation of the output of a system (needs lots more memory and calculating time) compared with the difference equation.

Convolution is *commutative* which means that it is also true to write:

$$\boxed{y[n]=\sum_{i=0}^{\infty}h[i]x[n-i]} \tag{1B.31}$$

Discrete-time convolution can be illustrated as follows. Suppose the unit-pulse response is that of a filter of finite length *k*. Then the output of such a filter is:

$$\begin{aligned} y[n] &= h[n] * x[n] \\ &= \sum_{i=0}^{\infty} h[i] x[n-i] \\ &= h[0]x[n] + h[1]x[n-1] + \ldots + h[k]x[n-k] \end{aligned} \tag{1B.32}$$

Graphically, this summation can be viewed as two buffers, or arrays, sliding past one another. The array locations that overlap are multiplied and summed to form the output at that instant.

Graphical view of the convolution operation in discrete-time

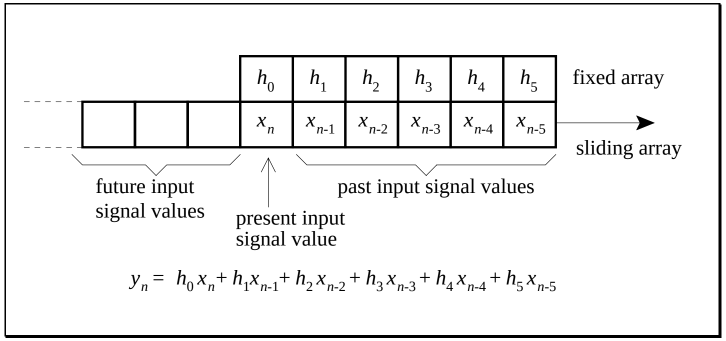


Figure 1B.4

In other words, the output at time *n* is equal to a linear combination of past and present values of the input signal, *x*. The system can be considered to have a memory because at any particular time, the output is still responding to an input at a previous time.

Discrete-time convolution can be implemented by a *transversal digital filter*:

Transversal digital filter performs discrete-time convolution

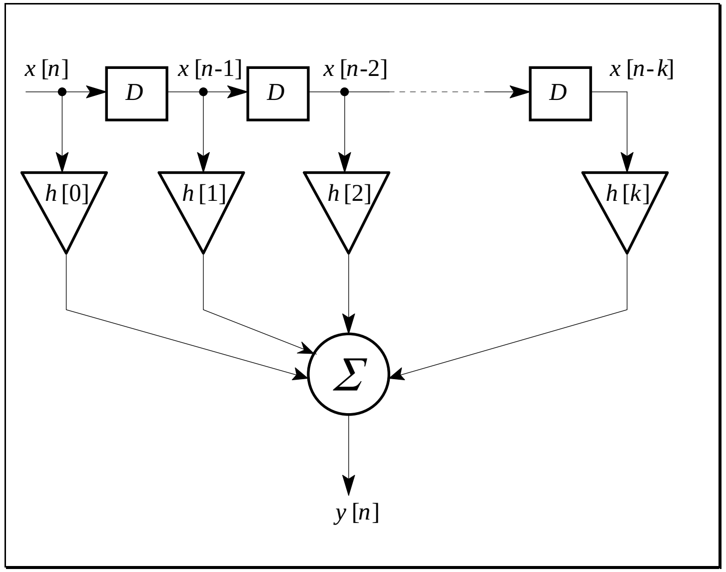


Figure 1B.5

MATLAB® can do convolution for us. Use the `conv` *function.*

System Memory

A system's memory can be roughly interpreted as a measure of how significant past inputs are on the current output. Consider the two unit-pulse responses below:

System memory depends on the unit-pulse response…

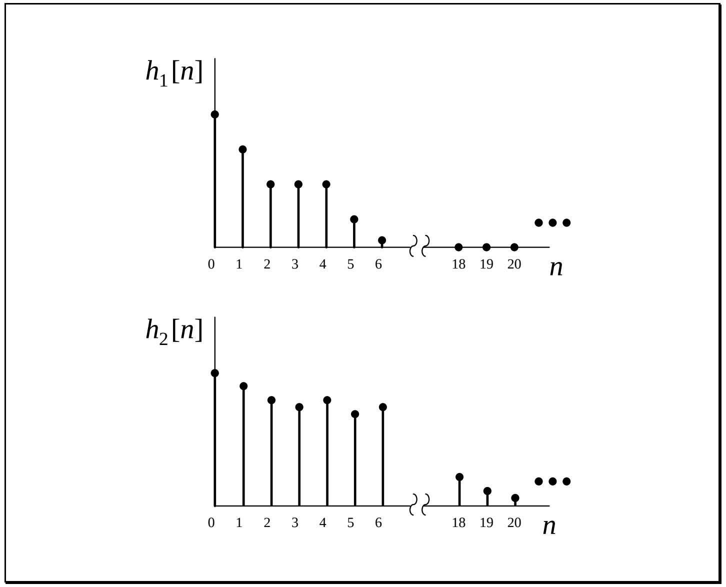


Figure 1B.6

System 1 depends strongly on inputs applied five or six iterations ago and less so on inputs applied more than six iterations ago. The output of system 2 depends strongly on inputs 20 or more iterations ago. System 1 is said to have a shorter memory than system 2.

It is apparent that a measure of system memory is obtained by noting how quickly the system unit-pulse response decays to zero: *the more quickly a system's weighting sequence goes to zero, the shorter the memory*. Some applications require a short memory, where the output is more readily influenced by the most recent behaviour of the input signal. Such systems are *fast responding*. A system with long memory does not respond as readily to changes in the recent behaviour of the input signal and is said to be *sluggish*.

…specifically - on how long it takes to decay to zero.

System Stability

$$\boxed{\text{A system is stable if its output signal remains bounded in response to any bounded signal.}} \tag{1B.33}$$

BIBO stability defined

If a bounded input (BI) produces a bounded output (BO), then the system is termed BIBO stable. This implies that:

$$\lim_{i \to \infty} h[i] = 0 \tag{1B.34}$$

This is something not readily apparent from the difference equation. A more thorough treatment of system stability will be given later.

What can you say about the stability of the system described by Eq. (1B.21)?

Convolution in Linear Time-invariant Continuous-time Systems

The input / output relationship of a continuous time system can be specified in terms of a convolution operation between the input and the impulse response of the system.

Deriving convolution for the continuous-time case

Recall that we can consider the impulse as the limit of a rectangle function:

$$x_r(t) = \frac{1}{T}\operatorname{rect}\left(\frac{t}{T}\right) \tag{1B.35}$$

Start with a rectangle input

as $T \to 0$. The system response to this input is:

$$y(t) = y_r(t) \tag{1B.36}$$

and the output response.

and since:

$$\lim_{T \to 0} x_r(t) = \delta(t) \tag{1B.37}$$

As the input approaches an impulse function then:

then the output approaches the impulse response

$$\lim_{T\to 0} y_r(t) = h(t) \tag{1B.38}$$

Now expressing the general input signal as the limit of a staircase approximation as shown in Figure 1B.7:

Treat an arbitrary input waveform as a sum of rectangles

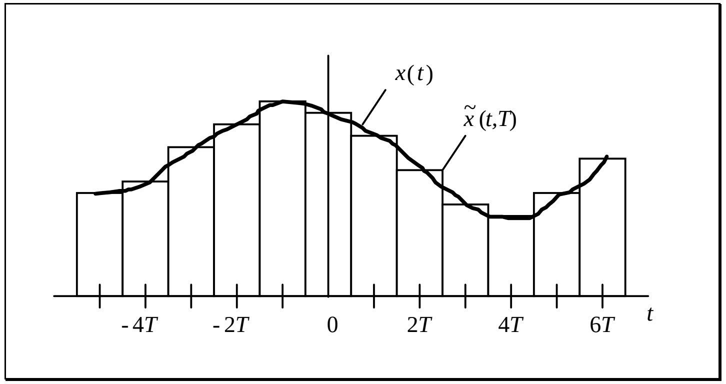


Figure 1B.7

we have:

which get smaller and smaller and eventually approach the original waveform

$$x(t) = \lim_{T\to 0} \tilde{x}(t,T) \tag{1B.39}$$

where:

$$\tilde{x}(t,T) = \sum_{i=-\infty}^{\infty} x(iT)\,\mathrm{rect}\left(\frac{t-iT}{T}\right) \tag{1B.40}$$

We can rewrite Eq. (1B.40) using Eq. (1B.35) as:

The staircase is just a sum of weighted rectangle inputs…

$$\tilde{x}(t,T) = \sum_{i=-\infty}^{\infty} x(iT)Tx_r(t-iT) \tag{1B.41}$$

Since the system is time-invariant, the response to $x_r(t-iT)$ is $y_r(t-iT)$. Therefore the system response to $\tilde{x}(t,T)$ is:

$$\tilde{y}(t,T) = \sum_{i=-\infty}^{\infty} x(iT)Ty_r(t-iT) \tag{1B.42}$$

…and we already know the output…

because superposition holds for linear systems. The system response to $x(t)$ is just the response:

$$y(t) = \lim_{T\to 0} \tilde{y}(t,T) = \lim_{T\to 0} \sum_{i=-\infty}^{\infty} x(iT)y_r(t-iT)T \tag{1B.43}$$

…even in the limit as the staircase approximation approaches the original input

When we perform the limit, $x(iT)\to x(\tau)$, $y_r(t-iT)\to h(t-\tau)$ and $T\to d\tau$. Hence the output response can be expressed in the form:

$$\boxed{y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau} \tag{1B.44}$$

Convolution integral for continuous-time systems defined

If the input $x(t)=0$ for all $t<0$ then:

$$y(t) = \int_{0}^{\infty} x(\tau)h(t-\tau)d\tau \tag{1B.45}$$

Convolution integral if the input starts at time t=0

If the input is causal, then $h(t-\tau)=0$ for negative arguments, i.e. when $\tau > t$. The upper limit in the integration can then be changed so that:

$$y(t) = \int_{0}^{t} x(\tau)h(t-\tau)d\tau \tag{1B.46}$$

Convolution integral if the input starts at time t=0, and the system is causal

Once again, it can be shown that convolution is *commutative* which means that it is also true to write (c*ompare with Eq. (1B.31)*):

$$\boxed{y(t) = \int_{0}^{t} h(\tau)x(t-\tau)d\tau} \tag{1B.47}$$

Alternative way of writing the convolution integral

With the convolution operation denoted by an asterisk, “*”, the input / output relationship becomes:

Convolution notation for a continuous-time system

$$\boxed{y(t) = h(t) * x(t)} \tag{1B.48}$$

Graphically, we can represent the system as:

Graphical notation for a continuous-time system using the impulse response

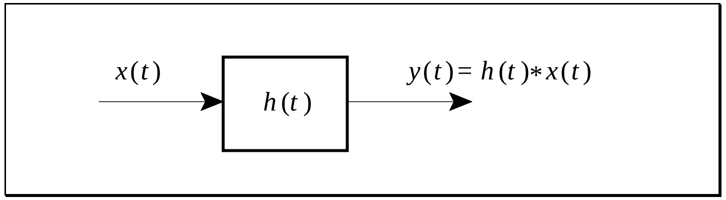


Figure 1B.8

It should be pointed out, once again, that the convolution relationship is *only valid when there is no initial energy stored in the system. ie. initial conditions are zero. The output response using convolution is just the ZSR.*

Graphical Description of Convolution

Consider the following continuous-time example which has a *causal* impulse response function. A causal impulse response implies that there is no response from the system until an impulse is applied at $t = 0$. In other words, $h(t) = 0$ for $t < 0$. Let the impulse response of the system be a decaying exponential, and let the input signal be the unit-step:

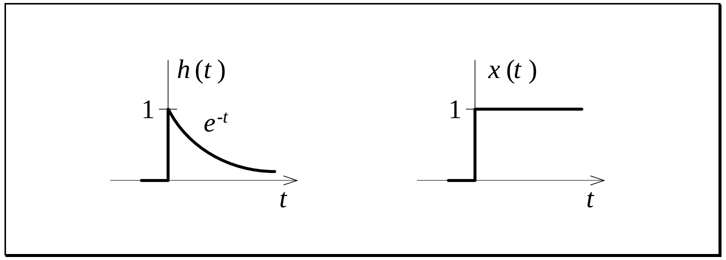


Figure 1B.9

Using graphical convolution, the output $y(t)$ can be obtained. First, the input signal is flipped in time about the origin. Then, as the time “parameter” t advances, the input signal “slides” past the impulse response – in much the same way as the input values slide past the unit-pulse values for discrete-time convolution. You can think of this graphical technique as the continuous-time version of a digital transversal filter (you might like to think of it as a discrete-time system and input signal, with the time delay between successive values so tiny that the finite summation of Eq. (1B.30) turns into a continuous-time integration).

When $t = 0$, there is obviously no overlap between the impulse response and input signal. The output must be zero since we have assumed the system to be in the zero-state (all initial conditions zero). Therefore $y(0) = 0$. This is illustrated below:

Graphical illustration of continuous-time - “snapshot” at t=0

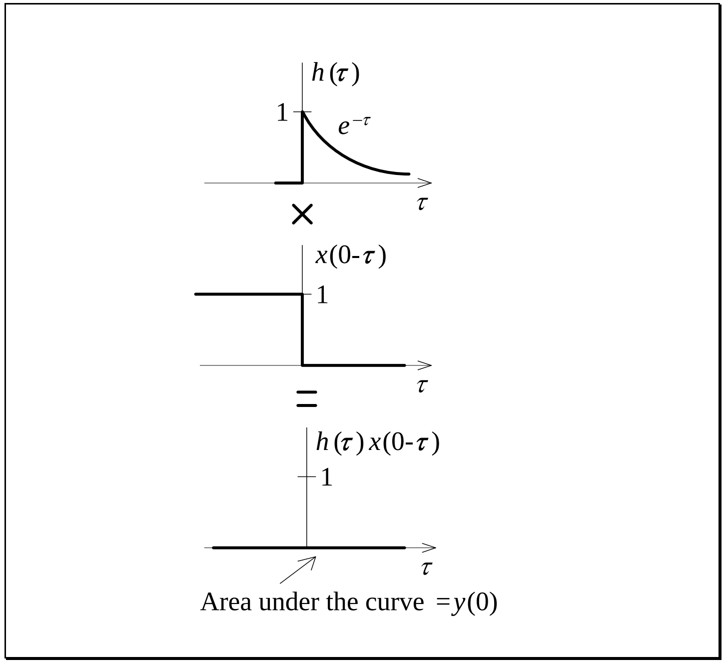


Figure 1B.10

Letting time “roll-on” a bit further, we take a snapshot of the situation when $t=1$. This is shown below:

Graphical illustration of continuous-time - “snapshot” at t=1

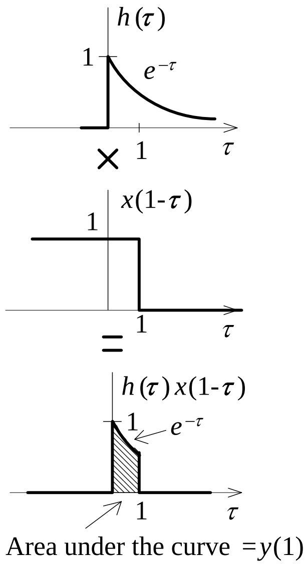


Figure 1B.11

The output *value* at $t=1$ is now given by:

$$\begin{aligned} y(1) &= \int_0^1 h(\tau)x(1-\tau)d\tau \\ &= \int_0^1 e^{-\tau}d\tau = \left[e^{-\tau}\right]_0^1 = 1-e^{-1} \approx 0.63 \end{aligned} \tag{1B.49}$$

Taking a snapshot at $t=2$ gives:

Graphical illustration of continuous-time - “snapshot” at t=2

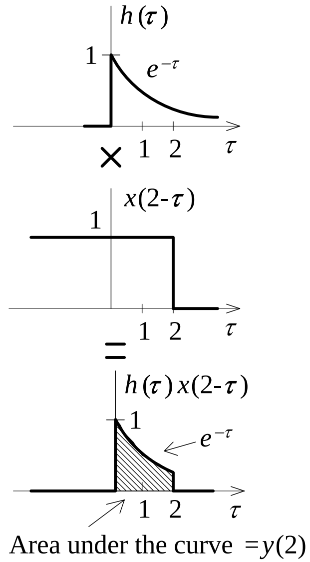


Figure 1B.12

The output *value* at $t=2$ is now given by:

$$\begin{aligned} y(2) &= \int_0^2 h(\tau)x(2-\tau)d\tau \\ &= \int_0^2 e^{-\tau}d\tau = \left[e^{-\tau}\right]_0^2 = 1-e^{-2} \approx 0.86 \end{aligned} \tag{1B.50}$$

If we keep evaluating the output for various values of t, we can build up a graphical picture of the output for all time:

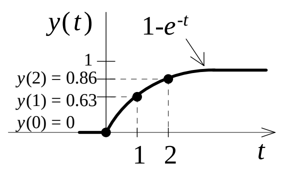


Figure 1B.13

In this simple case, it is easy to verify the graphical solution using Eq. (1B.47). The output value at any time t is given by:

$$\begin{aligned} y(t) &= \int_0^t h(\tau)x(t-\tau)d\tau \\ &= \int_0^t e^{-\tau}d\tau = \left[e^{-\tau}\right]_0^t = 1-e^{-t} \end{aligned} \tag{1B.51}$$

In more complicated situations, it is often the graphical approach that provides a quick insight into the form of the output signal, and it can be used to give a rough sketch of the output without too much work.

Properties of Convolution

In the following list of continuous-time properties, the notation $x(t) \rightarrow y(t)$ should be read as "the input $x(t)$ produces the output $y(t)$". Similar properties also hold for discrete-time convolution.

Convolution properties

$$ax(t) \rightarrow ay(t) \tag{1B.52a}$$

$$x_1(t)+x_2(t) \rightarrow y_1(t)+y_2(t) \tag{1B.52b}$$

$$a_1x_1(t)+a_2x_2(t) \rightarrow a_1y_1(t)+a_2y_2(t) \tag{1B.52c}$$

Linearity

$$x(t-t_0) \rightarrow y(t-t_0) \tag{1B.52d}$$

Time-invariance

Convolution is also associative, commutative and distributive with addition, all due to the linearity property.

Numerical Convolution

We have already looked at how to discretize a continuous-time system by discretizing a system's input / output differential equation. The following procedure provides another method for discretizing a continuous-time system. It should be noted that the two different methods produce two *different* discrete-time representations.

Computers work with discrete data

We start by thinking about how to simulate a continuous-time convolution with a computer, which operates on discrete-time data. The integral in Eq. (1B.47) can be discretized by setting $t = nT$:

$$y(nT) = \int_0^{nT} h(\tau)x(nT-\tau)d\tau \tag{1B.53}$$

By effectively reversing the procedure in arriving at Eq. (1B.47), we can break this integral into regions of width T:

$$\begin{aligned} y(nT) &= \int_0^T h(\tau)x(T-\tau)d\tau \\ &+ \int_T^{2T} h(\tau)x(2T-\tau)d\tau + \ldots \\ &+ \int_{iT}^{(i+1)T} h(\tau)x((i+1)T-\tau)d\tau + \ldots \end{aligned} \tag{1B.54}$$

which can be rewritten using the summation symbol:

$$y(nT) \approx \sum_{i=0}^{n} \int_{iT}^{iT+T} h(\tau)x(nT-\tau)d\tau \tag{1B.55}$$

If T is small enough, $h(\tau)$ and $x(nT-\tau)$ can be taken to be constant over each interval:

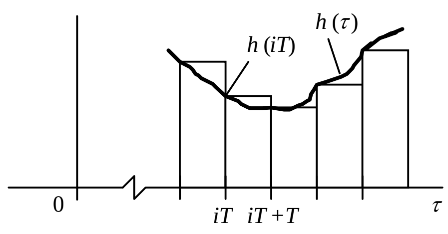


Figure 1B.14

That is, apply Euler's approximation:

$$\begin{aligned} h(\tau) &\approx h(iT) \\ x(nT-\tau) &\approx x(nT-iT) \end{aligned} \tag{1B.56}$$

so that Eq. (1B.55) becomes:

$$y(nT) \approx \sum_{i=0}^{n} \int_{iT}^{iT+T} h(iT)x(nT-iT)d\tau \tag{1B.57}$$

Since the integrand is constant with respect to τ, it can be moved outside the integral which is easily evaluated:

$$y(nT) \approx \sum_{i=0}^{n} h(iT)x(nT-iT)T \tag{1B.58}$$

We approximate the integral with a summation

Writing in the notation for discrete-time signals, we have the following input / output relationship:

$$\boxed{y[n] \approx \sum_{i=0}^{n} h[i]x[n-i]T, \quad n = 0, 1, 2, \ldots} \tag{1B.59}$$

Convolution approximation for causal systems with inputs applied at t=0

This equation can be viewed as the convolution-summation representation of a linear time-invariant system with unit-pulse response $Th[n]$, where $h[n]$ is the sampled version of the impulse response $h(t)$ of the original continuous-time system.

Convolution with an Impulse

One very important particular case of convolution that we will use all the time is that of convolving a function with a delayed impulse. We can tackle the problem three ways: graphically, algebraically, or by using the concept that a system performs convolution. Using this last approach, we can surmise what the solution is by recognising that the convolution of a function $h(t)$ with an impulse is equivalent to applying an impulse to a system that has an impulse response given by $h(t)$:

Applying an impulse to a system creates the impulse response

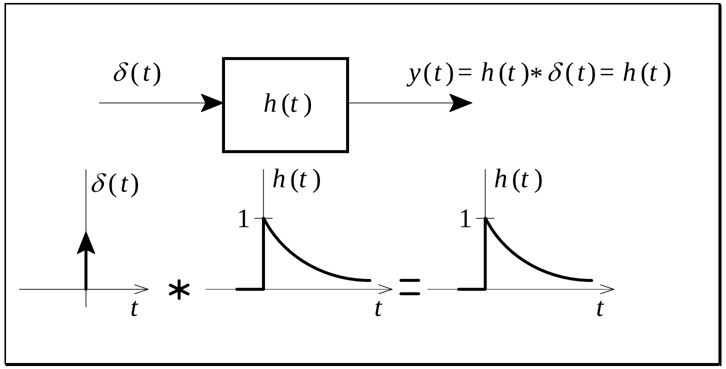


Figure 1B.15

The output, by definition, is the impulse response, $h(t)$. We can also arrive at this result algebraically by performing the convolution integral, and noting that it is really a sifting integral:

$$\delta(t) * h(t) = \int_{-\infty}^{\infty} \delta(\tau) h(t-\tau) d\tau = h(t) \tag{1B.60}$$

If we now apply a delayed impulse to the system, and since the system is time-invariant, we should get out a delayed impulse response:

Applying a delayed impulse to a system creates a delayed impulse response

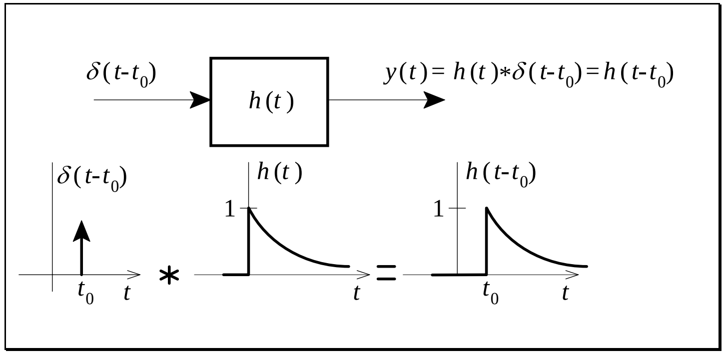


Figure 1B.16

Again, using the definition of the convolution integral and the sifting property of the impulse, we can arrive at the result algebraically:

$$\begin{aligned} \delta(t-t_0) * h(t) &= \int_{-\infty}^{\infty} \delta(\tau - t_0) h(t-\tau) d\tau \\ &= h(t-t_0) \end{aligned} \tag{1B.61}$$

Therefore, in general, we have:

$$\boxed{f(x) * \delta(x-x_0) = f(x-x_0)} \tag{1B.62}$$

This can be represented graphically as:

Convolving a function with an impulse shifts the original function to the impulse's location

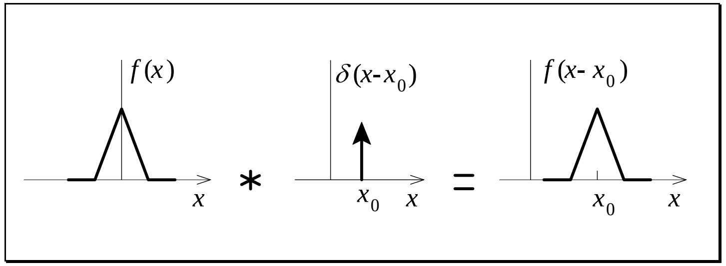


Figure 1B.17

Summary

- Systems are predominantly described by differential or difference equations – they are the equations of *dynamics*, and tell us how outputs and various *states* of the system change with time for a given input.
- Most systems can be derived from simple cause / effect relationships, together with a few conservation laws.
- Discrete-time signals occur naturally and frequently – they are signals that exist only at discrete points in time. Discrete-time systems are commonly implemented using microprocessors.
- We can approximate continuous-time systems with discrete-time systems by a process known as discretization – we replace differentials with differences.
- Convolution is another (equivalent) way of representing an input / output relationship of a system. It shows us features of the system that were otherwise "hidden" when written in terms of a differential or difference equation.
- Convolution introduces us to the concept of an impulse response for a continuous-time system, and a unit-pulse response for a discrete-time system. Knowing this response, we can determine the output for *any* input, *if the initial conditions are zero*.
- A system is BIBO stable if its impulse response decays to zero in the continuous-time case, or if its unit-pulse response decays to zero in the discrete-time case.
- Convolving a function with an impulse shifts the original function to the impulse's location.

References

Kamen, E. & Heck, B.: *Fundamentals of Signals and Systems using MATLAB*®, Prentice-Hall International, Inc., 1997.

Exercises

1.

The following continuous-time functions are to be uniformly sampled. Plot the discrete signals which result if the sampling period T is (i) $T = 0.1\text{ s}$, (ii) $T = 0.3\text{ s}$, (iii) $T = 0.5\text{ s}$, (iv) $T = 1\text{ s}$. How does the sampling time affect the accuracy of the resulting signal?

(a) $x(t) = 1$ (b) $x(t) = \cos 4\pi t$ (c) $x(t) = \cos 10\pi t$

2.

Plot the sequences given by:

(a) $y_1[n] = 3\delta[n+1] - \delta[n] + 2\delta[n-1] + 1/2\,\delta[n-2]$

(b) $y_2[n] = -4\delta[n] - \delta[n-2] + 3\delta[n-3]$

3.

From your solution in Question 2, find $a[n] = y_1[n] - y_2[n]$. Show graphically that the resulting sequence is equivalent to the sum of the following delayed unit-step sequences:

$$a[n] = 3u[n+1] - u[n-1] - 1/2\,u[n-2] - 9/2\,u[n-3] + 3u[n-4]$$

4.

Find $y[n] = y_1[n] + y_2[n]$ when:

$$y_1[n] = \begin{cases} 0, & n = -1, -2, -3, \ldots \\ (-1)^{n^2-1}, & n = 0, 1, 2, \ldots \end{cases}$$

$$y_2[n] = \begin{cases} 0, & n = -1, -2, -3, \ldots \\ 1/2\left(1 + (-1)^n\right), & n = 0, 1, 2, \ldots \end{cases}$$

5.

The following series of numbers is known as the Fibonacci sequence:

$$0,\ 1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34\ldots$$

(a) Find a difference equation which describes this number sequence $y[n]$, when $y[0]=0$.

(b) By evaluating the first few terms show that the following formula also describes the numbers in the Fibonacci sequence:

$$y[n]=\frac{1}{\sqrt{5}}\left[\left(0.5+\sqrt{1.25}\right)^{n}-\left(0.5-\sqrt{1.25}\right)^{n}\right]$$

(c) Using your answer in (a) find $y[20]$ and $y[25]$. Check your results using the equation in (b). Which approach is easier?

6.

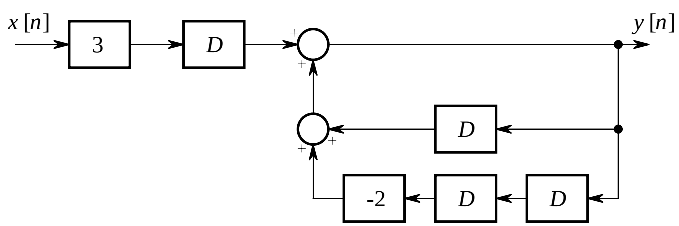
Construct block diagrams for the following difference equations:

(i) $y[n]=y[n-2]+x[n]+x[n-1]$

(ii) $y[n]=2y[n-1]-y[n-2]+3x[n-4]$

7.

(i) Construct a difference equation from the following block diagram:



(ii) From your solution calculate $y[n]$ for n = 0, 1, 2 and 3 given $y[-2]=-2$, $y[-1]=-1$, $x[n]=0$ for $n<0$ and $x[n]=(-1)^{n}$ for n = 0, 1, 2 …

8.

(a) Find the unit-pulse response of the linear systems given by the following equations:

(i) $y[n]=\frac{T}{2}\left(x[n]+x[n-1]\right)+y[n-1]$

(ii) $y[n]=x[n]-0.75x[n-1]+0.5y[n-1]$

(b) Determine the first five terms of the response of the equation in (ii) to the input:

$$x[n]=\begin{cases}0, & n=-2,-3,-4,\ldots\\ 1, & n=-1\\ (-1)^{n}, & n=0,1,2,\ldots\end{cases}$$

using (i) the basic difference equation, (ii) graphical convolution and (iii) the convolution summation. (Note $y[n]=0$ for $n\le -2$).

9.

For the single input-single output continuous- and discrete-time systems characterized by the following equations, determine which coefficients must be zero for the systems to be

(a) linear

(b) time invariant

(i) $a_1\left(\frac{d^3y}{dt^3}\right)^2+a_2\frac{d^2y}{dt^2}+\left(a_3+a_4y+a_5\sin t\right)\frac{dy}{dt}+a_6y=a_7x$

(ii) $a_1y^2[n+3]+a_2y[n+2]+\left(a_3+a_4y[n]+a_5\sin(n)\right)y[n+1]+a_6y[n]=a_7x[n]$

10.

To demonstrate that nonlinear systems do not obey the principle of superposition, determine the first five terms of the response of the system:

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

to the input:

$$x_1[n] = \begin{cases} 0, & n = -1, -2, -3, \ldots \\ 1, & n = 0, 1, 2, \ldots \end{cases}$$

If $y_1[n]$ denotes this response, show that the response of the system to the input $x[n] = 2x_1[n]$ is not $2y_1[n]$.

Can convolution methods be applied to nonlinear systems? Why?

11.

A system has the unit-pulse response:

$$h[n] = 2u[n] - u[n-2] - u[n-4]$$

Find the response of this system when the input is the sequence:

$$\delta[n] - \delta[n-1] + \delta[n-2] - \delta[n-3]$$

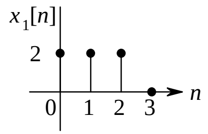
using (i) graphical convolution and (ii) convolution summation.

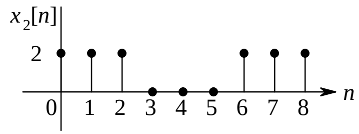
12.

For $x_1[n]$ and $x_2[n]$ as shown below find

(i) $x_1[n] * x_1[n]$ (ii) $x_1[n] * x_2[n]$ (iii) $x_2[n] * x_2[n]$

using (a) graphical convolution and (b) convolution summation.



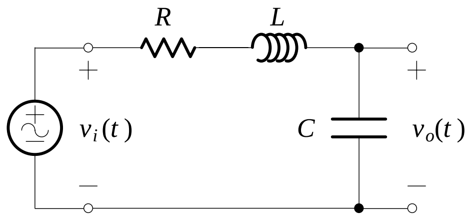


13.

Use MATLAB® and discretization to produce approximate solutions to the revision problem.

14.

Use MATLAB® to graph the output voltage of the following *RLC* circuit:

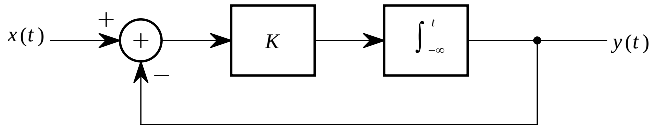


when $R = 2$, $L = C = 1$, $v_o(0) = 1$, $\dot{v}_o(0) = -1$ and $v_i(t) = \sin(t)u(t)$.

Compare with the exact solution: $v_o(t) = 0.5[(3+t)e^{-t} - \cos(t)]$, $t \geq 0$. How do you decide what value of T to use?

15.

A feedback control system is used to control a room's temperature with respect to a preset value. A simple model for this system is represented by the block diagram shown below:



In the model, the signal $x(t)$ represents the commanded temperature change from the preset value, $y(t)$ represents the produced temperature change, and t is measured in minutes. Find:

a) the differential equation relating $x(t)$ and $y(t)$,

b) the impulse response of the system, and

c) the temperature change produced by the system when the gain K is 0.5 and a step change of $0.75°$ is commanded at $t = 4\text{ min}$.

d) Plot the temperature change produced.

e) Use MATLAB® and numerical convolution to produce approximate solutions to this problem and compare with the theoretical answer.

16.

Use MATLAB® and the numerical convolution method to solve Q14.

17.

Quickly changing inputs to an aircraft rudder control are smoothed using a digital processor. That is, the control signal is converted to a discrete-time signal by an A/D converter, the discrete-time signal is smoothed with a discrete-time filter, and the smoothed discrete-time signal is converted to a continuous-time, smoothed, control signal by a D/A converter. The smoothing filter has the unit-pulse response:

$$h[nT] = \left(0.5^n - 0.25^n\right)u[nT], \quad T = 0.25\text{ s}$$

Find the zero-state response of the discrete-time filter when the input signal samples are:

$$x[nT] = \{1, 1, 1\}, \quad T = 0.25\text{ s}$$

Plot the input, unit-pulse response, and output for $-0.75 \le t \le 1.5\text{ s}$.

18.

A wave staff measures ocean wave height in meters as a function of time. The height signal is sampled at a rate of 5 samples per second. These samples form the discrete-time signal:

$$s[nT] = \cos(2\pi(0.2)nT + 1.1) + 0.5\cos(2\pi(0.3)nT + 1.5)$$

The signal is transmitted to a central wave-monitoring station. The transmission system corrupts the signal with additive noise given by the MATLAB® function:

```
function n0=drn(n)
  N=size(n,2);
  rand('seed', 0);
  no(1)=rand-0.5;
  for I=2:N;
    no(i)=0.2*no(i-1)+(rand-0.5);
end
```

The received signal plus noise, $x[nT]$, is processed with a low-pass filter to reduce the noise.

The filter unit-pulse response is:

$$h[nT] = \{0.182(0.76)^n - 0.144(0.87)^n \cos(0.41n) + 0.194(0.87)^n \sin(0.41n)\}u[nT]$$

Plot the sampled height signal, $s[nT]$, the filter input signal, $x[nT]$, the unit-pulse response of the filter, $h[nT]$, and the filter output signal $y[nT]$, for $0 \le t \le 6\text{ s}$.

Gustav Robert Kirchhoff (1824-1887)

Kirchhoff was born in Russia, and showed an early interest in mathematics. He studied at the University of Königsberg, and in 1845, while still a student, he pronounced Kirchhoff's Laws, which allow the calculation of current and voltage for *any* circuit. They are the Laws electrical engineers apply on a routine basis – they even apply to non-linear circuits such as those containing semiconductors, or distributed parameter circuits such as microwave striplines.

He graduated from university in 1847 and received a scholarship to study in Paris, but the revolutions of 1848 intervened. Instead, he moved to Berlin where he met and formed a close friendship with Robert Bunsen, the inorganic chemist and physicist who popularized use of the "Bunsen burner".

In 1857 Kirchhoff extended the work done by the German physicist Georg Simon Ohm, by describing charge flow in three dimensions. He also analysed circuits using topology. In further studies, he offered a general theory of how electricity is conducted. He based his calculations on experimental results which determine a constant for the speed of the propagation of electric charge. Kirchhoff noted that this constant is approximately the speed of light – but the greater implications of this fact escaped him. It remained for James Clerk Maxwell to propose that light belongs to the electromagnetic spectrum.

Kirchhoff's most significant work, from 1859 to 1862, involved his close collaboration with Bunsen. Bunsen was in his laboratory, analysing various salts that impart specific colours to a flame when burned. Bunsen was using coloured glasses to view the flame. When Kirchhoff visited the laboratory, he suggested that a better analysis might be achieved by passing the light from the flame through a prism. The value of spectroscopy became immediately clear. Each element and compound showed a spectrum as unique as any fingerprint, which could be viewed, measured, recorded and compared.

Spectral analysis, Kirchhoff and Bunsen wrote not long afterward, promises "the chemical exploration of a domain which up till now has been completely closed." They not only analysed the known elements, they discovered new ones. Analyzing salts from evaporated mineral water, Kirchhoff and Bunsen detected a blue spectral line – it belonged to an element they christened *caesium* (from the Latin *caesius*, sky blue). Studying lepidolite (a lithium-based mica) in 1861, Bunsen found an alkali metal he called rubidium (from the Latin *rubidius*, deepest red). Both of these elements are used today in atomic clocks. Using spectroscopy, ten more new elements were discovered before the end of the century, and the field had expanded enormously – between 1900 and 1912 a "handbook" of spectroscopy was published by Kayser in six volumes comprising five thousand pages!

"[Kirchhoff is] a perfect example of the true German investigator. To search after truth in its purest shape and to give utterance with almost an abstract self-forgetfulness, was the religion and purpose of his life." – Robert von Helmholtz, 1890.

Kirchhoff's work on spectrum analysis led on to a study of the composition of light from the Sun. He was the first to explain the dark lines (Fraunhofer lines) in the Sun's spectrum as caused by absorption of particular wavelengths as the light passes through a gas. Kirchhoff wrote "It is plausible that spectroscopy is also applicable to the solar atmosphere and the brighter fixed stars." We can now analyse the collective light of a hundred billion stars in a remote galaxy billions of light-years away – we can tell its composition, its age, and even how fast the galaxy is receding from us – simply by looking at its spectrum!

As a consequence of his work with Fraunhofer's lines, Kirchhoff developed a general theory of emission and radiation in terms of thermodynamics. It stated that a substances capacity to emit light is equivalent to its ability to absorb it at the same temperature. One of the problems that this new theory created was the "blackbody" problem, which was to plague physics for forty years. This fundamental quandary arose because heating a black body – such as a metal bar – causes it to give off heat and light. The spectral radiation, which depends only on the temperature and not on the material, could not be predicted by classical physics. In 1900 Max Planck solved the problem by discovering quanta, which had enormous implications for twentieth-century science.

In 1875 he was appointed to the chair of mathematical physics at Berlin and he ceased his experimental work. An accident-related disability meant he had to spend much of his life on crutches or in a wheelchair.