Indian Institute of Technology Bombay

Dept of Electrical Engineering

Handout 2 EE 210 Signals and Systems

Lecture Notes 1 July 22, 2015

Introduction

1 Signals

“ A collection of *sensible* objects ”

Merriam Webster (http://m-w.com) has the following info.

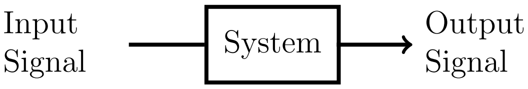
Main Entry:	sen·si·ble
Meaning	1: of a kind to be felt or perceived: as a: **perceptible to the senses or to reason or understanding**

It is not necessary that ‘sensible’ objects come from a physical sensing device. It can be a conceptual entity like an impulse, or an abstract collection in some space. Our aim is to develop this naive representation to concrete mathematical abstractions, simple enough to understand and manipulate, but sufficient to capture the *true* properties.

2 Systems

“ Meaningful operations on signals ”

Our discussion on signals apply here too. In other words, though *meaningful* and *sensible* are semantically alright, we need an approach which has more syntax, to create a precise (more formal) notion of signals and systems. Systems can be treated as a mapping from signals to signals. Pictorially,



System represents an input-output relationship.

In this representation, the terms **input** and **output** carry their conventional meaning, that we engineers are familiar with. While we will deal with *almost* any signal thrown at us, say by theory or practice, most of our discussions will consider only a particular class of systems, the so called **linear systems**[1]. As the name implies, linear systems perform linear operations of the input signals (examples: add two or more input signals, scale an input signal). Understanding linear systems is the back-bone of any signal processing course.

In order to be systematic, we will introduce some taxonomy on linear systems as well as the input signals. This is quite a natural thing to do, as our tools and techniques may differ among these classes, but not the underlying philosophy. Let us introduce some useful notations now.

[1] let us keep non-linearity for some other day

3 Notation

$\mathbb{N}$	:	Natural numbers
$\mathbb{Z}$	:	Integers
$\mathbb{Q}$	:	Rational Numbers
$\mathbb{R}$	:	Real Numbers
$\mathbb{C}$	:	Complex Numbers
$f(\cdot)$	:	function from $\mathbb{R} \to \mathbb{C}$
$\hat{f}(\cdot)$	:	function from $\mathbb{C} \to \mathbb{C}$
$\mathcal{C}^k$	:	Class of k-times continuously differentiable functions
$\mathbb{R}_+$	:	Non-negative real numbers

We will denote a closed interval by square brackets, i.e. $[a, b]$ for the interval from point a to point b in the real axis. Similarly (a, b) will denote the open interval.

Since many of you come from diverse backgrounds, it is instructive to have an example in mind which we all can visualize. We will introduce the concept of interpolation to sketch a system level view of signal manipulation. This should reassure the newcomers to the field that every aspect we learn has great practical impact. At times down the line, we may spent considerable time in seemingly mundane mathematical aspects, but it will be followed by seasons where we apply the theory to real-life problems. Once mastered, the beauty of the whole machinery will be greatly appreciated.

4 Interpolation: A starting example

Suppose we are given a set of measurements, say of temperature at different times for the month of June. There were 10 measurements every day for a total of 300 values. Assuming that there were no drastic variations [2], can we make a chart/plot of the temperature variations for the full month of June.

Before jumping to an answer think that the interval of time in consideration is of length $30 \times 24 \times 60 \times 60 = 2592000 sec$. We have just 300 values!. Assume for simplicity that the measuring times are evenly distributed on the time interval. Similar problems occur in many other areas, say in a stock market. However, the time-scales of interest and data-variations may be very different. Let us ask our question in a more formal manner.

Given,

- a set of $N+1$ indices t_i, $0 \le i \le N$.
- a function $f(\cdot)$ evaluated at these points, i.e. $f(t_i)$, $0 \le i \le N$

can we find $f(t)$ for the whole interval $t \in [t_0, t_N]$.

You will not take much time in figuring out that the answer is NO. There are so many answers that will fit the above description, i.e. problem is ill-posed given no other assumption about the function. However, the answer is YES if the function $f(\cdot)$ is assumed to be a polynomial of degree at most N. We can determine it completely from $N+1$ points. This then is the kind of philosophy that we can use in getting the in-between values.

[2] the word *drastic* is tricky and loose, we will make it precise later

4.1 Strategy I

Assume the unknown function $f(\cdot)$ was a polynomial of degree at most N. Given $f(t_i)$, $0 \le i \le N$, construct a polynomial of minimum degree $g(t)$ such that $g(t_i) = f(t_i), 0 \le i \le N$. The polynomial construction can be accomplished by the elegant interpolation formula by Lagrange. Define the **Lagrange polynomial**

$$L_i(t) = \prod_{j \neq i} \frac{t - t_j}{t_i - t_j}.$$

Notice that

$$L_i(t) = \begin{cases} 1 \text{ if } t = t_i \\ 0 \text{ if } t = t_j \,,\, j \neq i \end{cases} \tag{1}$$

Now, we can form $g(t)$ as

$$g(t) = \sum_i f(t_i) L_i(t). \tag{2}$$

Example 1. *Consider a function specified at three points as shown in Figure 1. Our aim is to find the interpolating polynomial of second degree.*

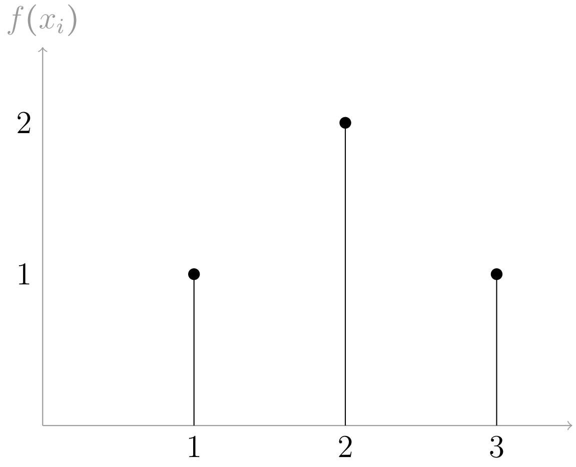


Figure 1: Samples for interpolation

Applying Lagrange interpolation formula,

$$\begin{aligned} g(x) &= \frac{(x-2)(x-3)}{2} - 2\frac{(x-1)(x-3)}{1} + \frac{(x-1)(x-2)}{2} && (3) \\ &= -x^2 + 4x - 2 && (4) \end{aligned}$$

See Figure 2 for a plot.

While Lagrange interpolation seems a natural step to do, it has several limitations. For example, in our problem of plotting temperatures, we may get a polynomial of degree as high as 300. While modern computers can perform such compuations, scalability will become an issue, particularly for images, videos etc. For many applications, we may need online interpolation, as data values keep coming in real time. Furthermore, many quantities of interest vary sufficiently slow, that fitting low degree polynomials between neighbouring values itself is sufficient. This last point is more like a greedy approach to interpolation, and let us in particular target the piece-wise polynomial approximation.

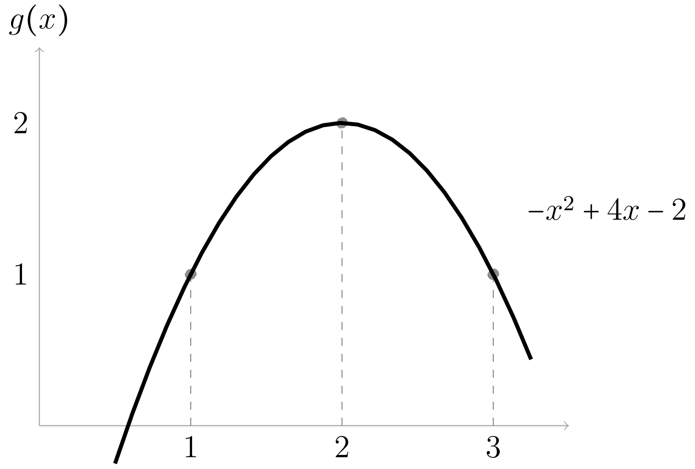


Figure 2: Interpolation in Example 1

4.2 Piece-wise polynomial approximation

The title suggests that we will stitch together pieces of polynomials. Each polynomial approximates the function well in some interval. Such polynomials over non-overlapping intervals are tailored to get the piece-wise polynomial approximation. We will call these intervals by the name **segments**. Every segment will use a low degree polynomial (eg: degree $0, 1, 3$ etc). To simplify the discussion (at the same time, without loosing much practical significance), we confine our discussion to the case where the segments are of equal length, i.e. successive samples are taken equi-distant in the $x-$axis.

4.2.1 Piece-wise Constant Approximation

This is best illustrated by an example.

Example 2. *Consider* 5 *points shown in the top of Figure 3. Let us construct an approximation for the interval* $[\frac{1}{2}, \frac{11}{2}]$.

Let the successive samples are apart by τ units. Take every sample point and draw a horizontal line of length τ with the given point as its center. This approximation is shown in the bottom half of Figure 3.

Apart from its simplicity, piece-wise constant approximations are of immense value in integration theory. As far as signal processing is concerned, piece-wise constants fall terribly short of quality. But our motivation of introducing this here is to learn another fundamental idea, that of **system representation**. In particular, we can visualize the samples entering a blackbox and interpolated values coming out of the box.

To see this, imagine each sample which enters a blackbox getting replaced by a rectangle of constant width and variable height. More precisely, an input sample at time t_0 of value $f(t_0)$ will be replaced by a rectangle of height $f(t_0)$, its base sitting on $x-$axis with the mid-point at t_0.

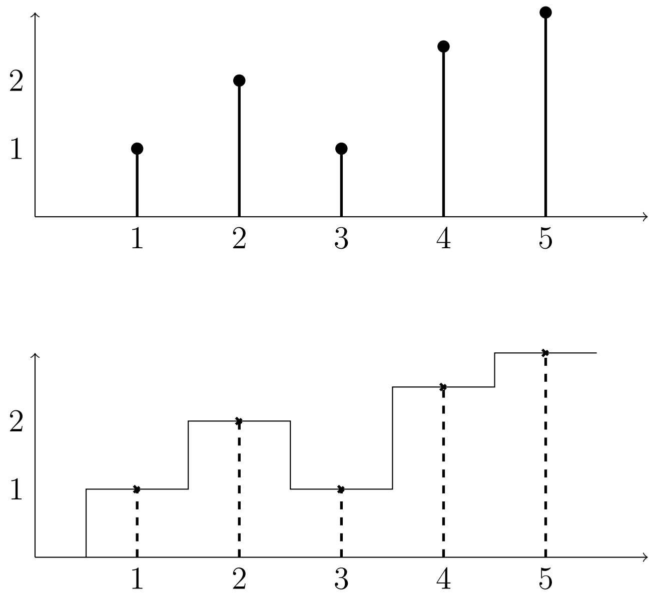


Figure 3: Input samples and their piece-wise constant approximation

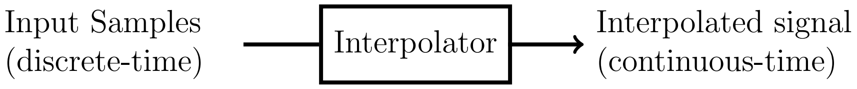


Figure 4: Piece-wise constant interpolation

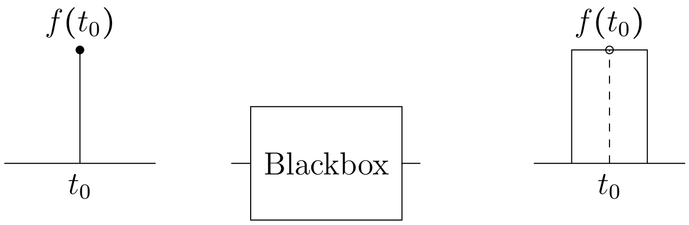


Figure 5: Replacing signal value by a rectangle

We can treat the rectangle with unit base and height as an **atom** (more formally it is called the interpolating **kernel**). The system is built using such atoms. Each input sample stretches the atomic height while keeping the base-width the same. The center of the atom coincides with the position of the sample. Notice further that the input height can be positive or negative or even complex, with the appropriate substitution clear from the context.

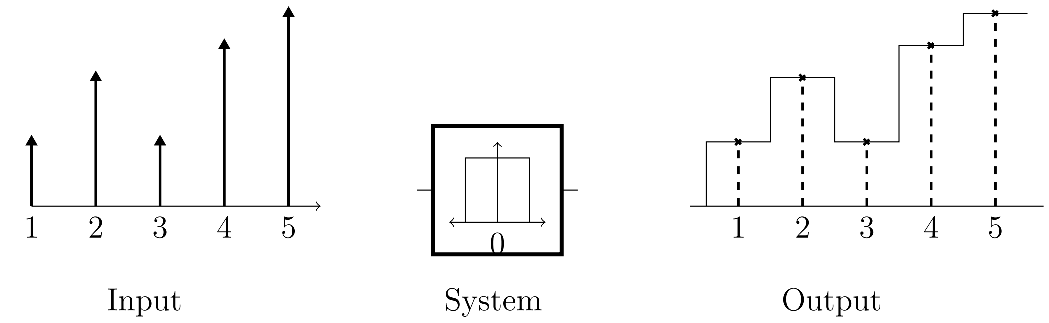


Figure 6: System representation of piece-wise constant interpolation

We can label the black-box by the picture of the atom. This will open new possibilities by using different atoms[3]. The piece-wise constant interpolation system is pictorially shown in Figure 6. The important take-home point here, in addition to the concept of kernel, is that the output waveform is the superposition of the effects of the individual samples entering the interpolator/system.

A careful reader might have noticed the arrows on top of the input samples in Figure 6. This is a symbolization. It tells us that each arrow is replaced by a scaled atom/kernal at the output of the system. A vertical line with an arrow on top symbolizes an impulse. A unit impulse at origin is also known as Dirac delta measure. Such representations are often known as symbolic calculus. We will encounter further details of this calculus later, known as 'symbolic calculus of impulses'.

4.2.2 Linear Interpolation

In signal processing parlance, piece-wise constant interpolation is not a sane thing to do. We may demand our perceptible signals to vary smoothly, while piece-wise constants fail to even draw a continuous figure. Our quest now is for a simple interpolator, which gives a continuous function at the output. This is easily obtained by connecting successive input samples by a straight line, see Figure 7. Such line based interpolation is known as linear interpolation.

Notice that we have extended our interpolation interval to $[0,6]$ by adding zero-valued samples at $t = 0$ and $t = 6$. This is just for a sensible termination, and will not have any impact on the interpolation for $t \in (1,5)$.

Let us ask the question whether linear interpolation admits an atomic superposition similar to the one we had for piece-wise constants. In particular, can we replace every sample by a scaled atomic diagram and get linearly interpolated values at the output using superposition. The answer is YES, but we need to pay attention to a small detail. The constructed atomic width is 2 units of the sample interval, as opposed to the single unit

[3] unlike chemistry, here we are not bothered about putting together an integer number of atoms.

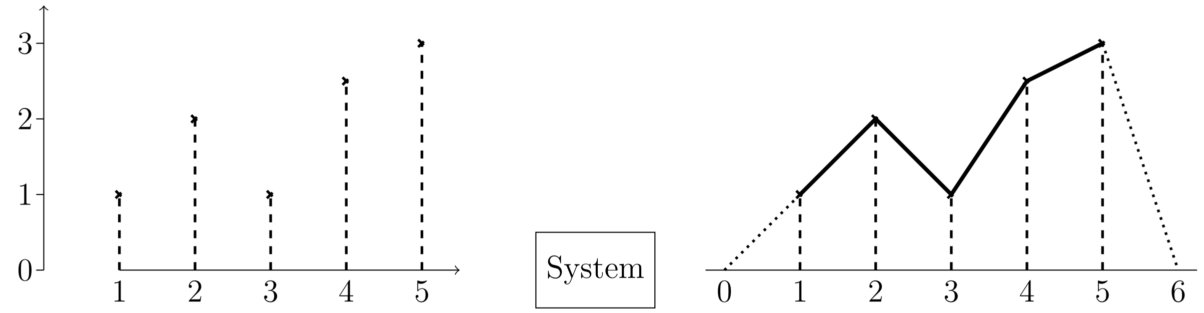


Figure 7: Linear interpolation

we had in piece-wise constants. In the above example, the base of the atomic function will occupy the interval $[-1, 1]$. This is mildly confusing, since successive atomic replacements will now overlap in time at the output, see the illustratation in figure. The overlap of

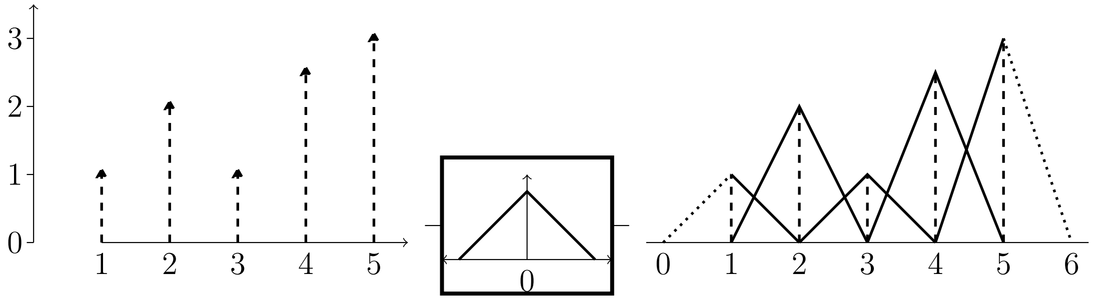


Figure 8: Atomic Composition of Linear interpolation

atomic-replacements will result in the super-position of these. In other words, we need to add the effect of all overlapping segments at any given time to get the total effect. Notice that at for any interval $(i, i+1), 1 \leq i \leq 4$ on the right side of 8, there are two overlapping lines. The sum of two lines is another line. Further, at each sample instant t_i the lines sum to $f(t_i)$. This uniquely identifies the superposed line, shown as the thick line in Figure 9.

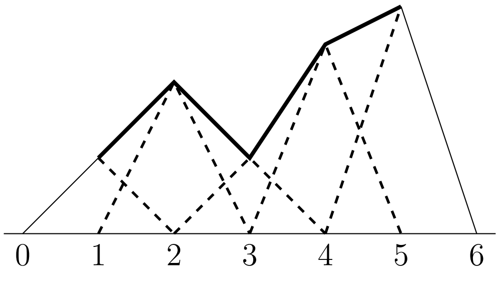


Figure 9: Linear interpolation and atomic superposition

4.2.3 Cubic Interpolation

Perhaps the most widely used interpolation is that of cubic interpolation. As the name shows, we will use third degree polynomials to construct an approximation to the orginal function $f(\cdot)$ using the given $N+1$ samples at locations $t_0, t_1, \cdot\cdot, t_N$. The conditions that we insist for the approximation function $g(t)$ are

1. $$f(t_i) = g(t_i)\,,\, 0 \le i \le N \tag{5}$$

2. $$f'(t_i) = g'(t_i)\,,\; 1 \le i \le N-1 \tag{6}$$

Let us find the cubic polynomial approximation between the points t_1 and t_2 with function values $f(t_1)$ and $f(t_2)$ respectively. Once we construct this, we can use the same technique to construct the cubic interpolation between any two points. The governing equations for any cubic polynomial $at^3 + bt^2 + ct + d$ satisfying (5) are,

$$at_1^3 + bt_1^2 + ct_1 + d = f(t_1) \tag{7}$$

$$at_2^3 + bt_2^2 + ct_2 + d = f(t_2) \tag{8}$$

We further need the right-derivative of $g(x)$ to be the same as the left-derivative at t_i, $1 \le i \le N-1$. Let us make the derivatives equal to β_1 and β_2 at t_1 and t_2 respectively. For the right-derivative at t_1 to be β_1,

$$3at_1^2 + 2bt_1 + c = \beta_1. \tag{9}$$

Similarly for getting a left-derivate of β_2 at t_2,

$$3at_2^2 + 2bt_2 + c = \beta_2. \tag{10}$$

By ensuring this derivative condition for the cubic polynomial that we find for every segment, we will indeed have $g(t)$ a differentiable function. A very common choice for β_i is the difference appromixation to the derivative. Specifically,

$$\beta_i = \frac{f(t_{i+1}) - f(t_{i-1})}{2}. \tag{11}$$

We thus have four equations (7) to (11) to evaluate the parameters a, b, c, d. The following properties will help us easily find the polynomial coefficients.

- w.l.o.g assume $t_1 = 0$, as we can port the design to any starting time by simply shifting the polynomial (drawing the picture from some t instead of 0).
- The interval length $t_2 - t_1$ w.l.o.g can be taken to be unity, since we already assumed that the sample points are equi-distant in time. This is justified, since the obtained interpolation for $[0,1]$ can be stretched/time-shortened to fit into $[0, t_2]$, and then use the previous property.

With $t_1 = 0$, $t_2 = 1$, $t_3 = 2$ and $t_{-1} = -1$, we can evaluate the polynomial coefficients.

$$a = -\frac{1}{2}f(-1) + \frac{3}{2}f(0) - \frac{3}{2}f(1) + \frac{1}{2}f(2) \tag{12}$$

$$b = f(-1) - \frac{5}{2}f(0) + 2f(1) - \frac{1}{2}f(2) \tag{13}$$

$$c = \frac{1}{2}f(1) - \frac{1}{2}f(-1) \tag{14}$$

$$d = f(0) \tag{15}$$

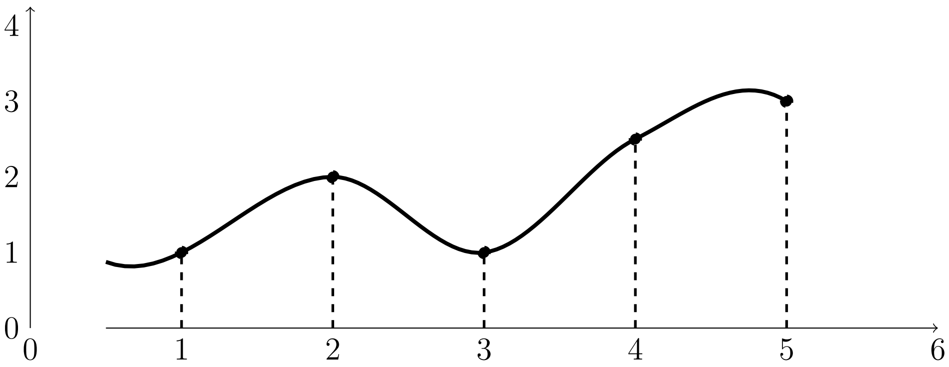


Figure 10: Cubic interpolation for $(1,1)\,(2,2)\,(3,1)\,(4,2.5)\,(5,3)$

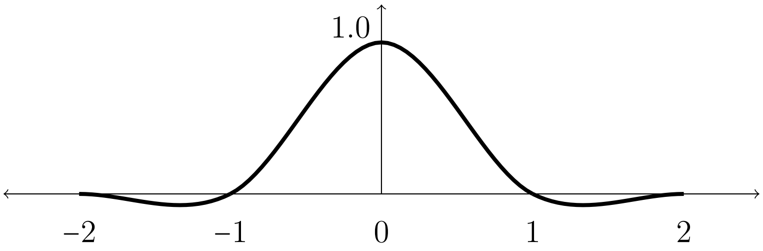


Figure 11: Output of a cubic interpolator to a single value

We now understand cubic interpolation, which has a lot of applications. However, our interests here is to use this model to say something more general about systems. For example, can we write the output of the interpolator as atomic superpositions, the same way we did for constant and linear interpolators. At first this looks a bit awkward, as the polynomials are of degree 3 and linearity is not visible in their very nature. Nevertheless, we indeed have such a representation, which is nothing but the output of the cubic interpolator to a Dirac delta measure, this is depicted in Figure 11.

Notice that we obtained the kernel by passing an impulse through the system. This generalizes, and we can find the response of many systems by passing an impulse through it. In this sense, the kernel function is also known as *impulse response.*

We can now show that the atomic superpositions of the kernel shown in Figure 11, when driven by the input samples, will indeed result in cubic interpolation. Without loss of generality, let us show this for the interpolation between points 0 and 1. Let f_{-1}, f_0, f_1, f_2 be the set of inputs and the interpolating polynomial be $p(t) = at^3 + bt^2 + ct + d$. Clearly,

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} & \frac{3}{2} & -\frac{3}{2} & \frac{1}{2} \\ 1 & -\frac{5}{2} & 2 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_{-1} \\ f_0 \\ f_1 \\ f_2 \end{bmatrix}. \tag{16}$$

With G as the matrix above, we can write (16) as $\bar{p} = G\bar{f}$, where the column vector $\bar{p}$ has the coefficients of polynomial $p(t)$, and $\bar{f}$ contains the input values. Take two vectors $\bar{\alpha}$ and $\bar{\beta}$ such that $\bar{f} = \bar{\alpha} + \bar{\beta}$. We will show that if we interpolate input $\bar{\alpha}$ and $\bar{\beta}$ separately and superpose, we will get $p(t)$, the interpolation of $\bar{f}$. In particular, if $\bar{\alpha}$ is the input sequence, our interpolation polynomial will have coefficients $G\bar{\alpha}$. Clearly,

$$G\bar{\alpha} + G\bar{\beta} = G(\bar{\alpha} + \beta) = G\bar{f}.$$

Thus, the sum of the polynomial coefficients for $\bar{\alpha}$ and $\bar{\beta}$ are same as the coefficients of $p(t)$. Since any discrete-time input can be written as a superposition of values at discrete-time instants, atomic superpositions are sufficient to get us cubic interpolation.

We studied interpolation to enable visualizing the superposition property. It can be hidden from a cursory look, as in the case of a cubic interpolation. More generally, how do we figure out that such representations exist? To our relief, it turns out that every **linear time-invariant system** (LTI) has such a property. What is an LTI system? How do we check whether a system is LTI?, please read on, the answers are ahead.

5 Class of Signals

Our structured approach is to divide the set of sensible objects (signals) and meaningful operations (systems) to classes. To see how classification helps, consider the collection of citizens of India. The operation we want: administer meaningful governance (say imparting basic education). We can define sets of people who have some properties in common (such as the same language), and define operations on each of these classes. The classification can also be based on convenience, observed facts or pedagogical reasons. Thus the partitions that we introduce are not that rigid.

5.1 Continuous-time and Discrete-time signals

A contiuous-time signal is defined for all $t \in \mathbb{R}$. One can interchange the term with functions $f(x), x \in \mathbb{R}$. Care should be applied that only the time index may be continuous, a continuous-time signal still can have discrete jumps. To emphasize, we will say that the signal itself is continuous when there are no jumps (see examples below).

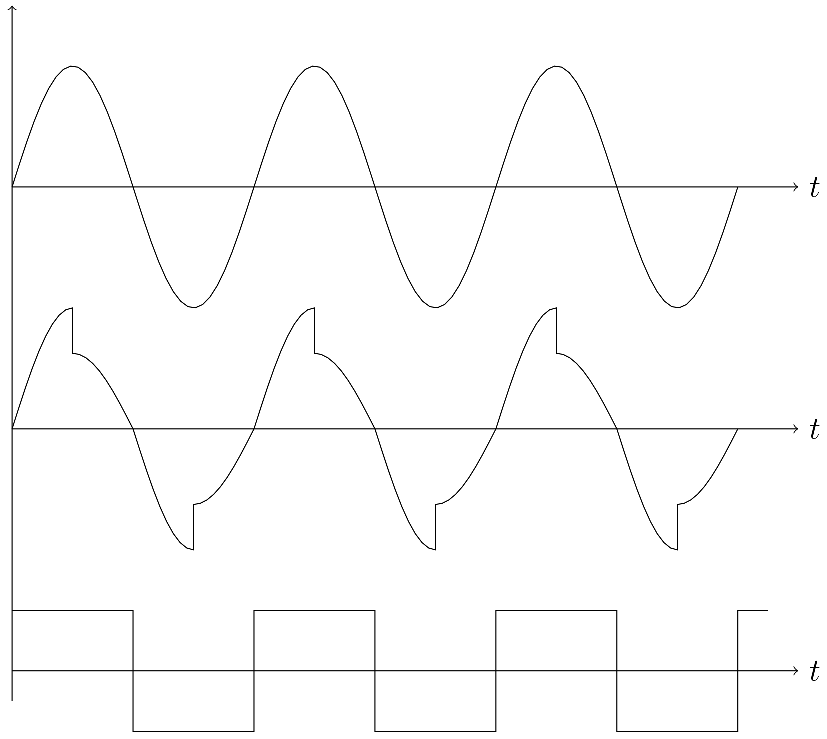


Figure 12: Examples of continuous-time signals

Example 3. *We denote the class of real-valued continuous signals/functions defined on the real axis by $\mathcal{C}^0$ (maybe you have forgotten the definition of continuity, please use wikipedia or any analysis textbook to update/refresh).*

Exercise 1. *Write down a meaningful operation for the class $\mathcal{C}^0$.*

Exercise 2. *Write down an operation which is not always meaningful in $\mathcal{C}^0$.*

Exercise 3. *Write down an operation which is never meaningful in $\mathcal{C}^0$.*

Our idea on *meaningful* (operation) is clearer now. We will improve further, and place this concept on sound logical and mathematical footing.

Stable Signals
A signal $f(t)$ is **stable** or **integrable** if,

$$\int_{\mathbb{R}} |f(t)| dt < \infty.$$

Not all signals are stable. For example, a periodic sinusoid is unstable by our definition. The notion of *local integrability* is central for the mathematical treatment of unstable signals. A signal $f(t)$ is locally integrable over the finite interval (a, b) if

$$\int_a^b |f(t)| dt < \infty.$$

The issue of stability apart, are there signals which are not time continuous?. In the digital age, the answer is more than YES. Almost everything that we do nowadays involves discrete/digital signals and their processing (eg: computer).

As opposed to continuous in time signals, **discrete-time** signals are defined on a set discrete set of time values.

The name discrete-time signal is actually a slight misnormer. The $x-$ axis can be treated simply as a set of indices. For example, in a still image, each index can correspond to the pixel position and the $y-$ axis can plot the corresponding intensity. Let us denote a discrete-time signal by $x[n]$ where n represents the index. See the following figure.

Example 4. *The operation,*

$$x[n] + x[n-1], \tag{17}$$

is meaningful when $x[n] \in \mathbb{R}, \forall n \in \mathbb{Z}$ [4]*.*

The discrete-time signals that we consider have a property that the interval between successive time instants is *visibly* an integer multiple of some real number, the largest of which we call the fundamental sampling interval (unofficial terminology) of x.

Exercise 4. *Does every discrete-time signal have a fundamental sampling interval. Write a proof for your answer.*

From now onwards, the discrete-time signals that we consider are assumed to have successive time-indices equi-distant. Thus index n happens at time nT, where T is the sampling interval.

The signals shown in Figure 1, the signal pattern repeats after constant intervals. On other hand, the well known Gaussian function is continuous and belongs to $\mathcal{C}^k, \forall k \in N$, but it does not repeat itself. Our second classification is based on the property of periodicity.

[4] recall that $\mathbb{R}, \mathbb{Z}$ and $\mathbb{N}$ stands for reals, integers and natural numbers respectively.

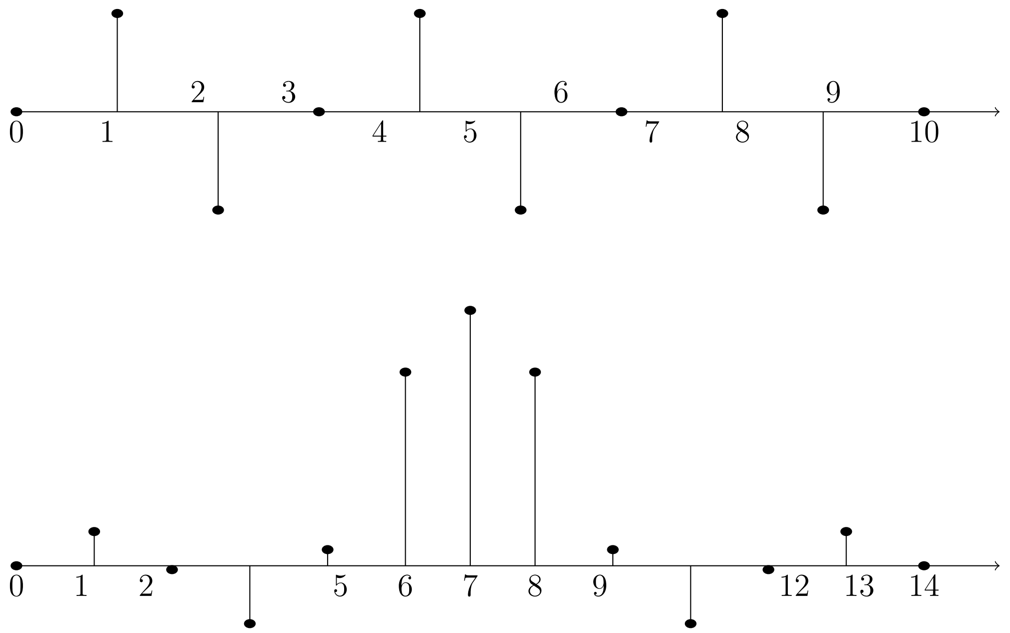


Figure 13: Examples of discrete-time signals

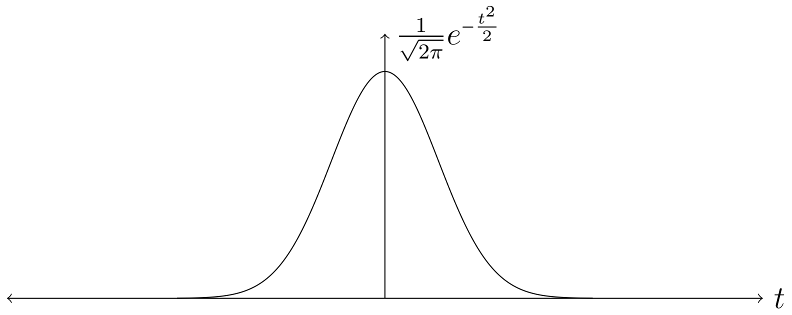


Figure 14: Gaussian Function

Exercise 5. *Show that*

$$\int_{\mathbb{R}} e^{-\frac{x^2}{2}} dx = \sqrt{2\pi}$$

5.2 Periodic and Non-periodic signals

This classification cuts across continuous time and discrete-time signals. We present the ideas for the continuous time case only. The discrete-time counterparts are easily obtained by extension, say by replacing integrals with summations and so on. A signal $f(\cdot)$ is considered **periodic** if there exists a real number T such that

$$f(t+T) = f(t) \ , \ \forall t \in \mathbb{R}.$$

We will say that f is $T-$periodic. Signals which do not have the above property is called non-periodic. Discrete-time signals can also be periodic or non-periodic. Most of the discrete-time signals which we encounter are non-periodic[5].

The AC (alternating current) signal coming through the power supply is a periodic signal, in fact a sinusoid of frequency 50Hz and root mean square voltage of 230Volts. We

[5] some may disagree, and argue that periodicity is indirectly used in most discrete-time operations

are naturally concerned with the amount of energy (or power) possessed by such a wave. For a continuous time signal $x(t)$, the energy is defined as

$$E_f = \int_{\mathbb{R}} |f(t)|^2 dt.$$

5.2.1 Energy Signals

If $E_f < \infty$, then the signal f is called a **square integrable** signal or **energy signal**.

Exercise 6. *Show that any bounded function with finite support is square integrable.*

5.2.2 Power Signals

Remember that a $T-$periodic waveform repeats itself over the entire time axis. Consequently, E_f is unbounded. On the other hand, the power, defined as energy per second can be computed as

$$P_f = \frac{1}{T} \int_0^T |f(t)|^2 dt.$$

A periodic signal with $P_f < \infty$ will be termed as a power signal. In general, if

$$\int_a^b |f(t)|^2 dt < \infty$$

for any finite real interval of the form (a, b), we will say that the signal is **locally square integrable**.

The properties that we described above covers most of the signal terminologies that we use in this course. Let us now look at some of the common system operations.

6 System Operations

The most common operations that we encounter in this course are scaling, shifting and addition of signals. In an abstract sense, a system (operation) can be represented by a deterministic mapping from the space of input signals to the space of output signals.

6.1 Scaling

The simplest scaling example is that of amplification.

Amplification of $f(t)$ by a real value a, defined as

$$g(t) = af(t), \forall t$$

is a meaningful operation.

Notice that the frequency of the waveform is unchanged by amplitude scaling. Yet another scaling operation, very common in wavelet analysis, is where the argument of the function is scaled, as opposed to the function itself. The argument scaling (time-scaling) will change the frequency components. See the following example for $f(t) = \sin(t)$ and $f(2t)$. The frequency of the time-scaled waveform is double that of the original, though there is no change in amplitude. Amplitude scaling is a **linear** operation, while time-scaling is non-linear. If you wonder what this means, a major part of our study is to make this differentiation, which will get more clear as we move on.

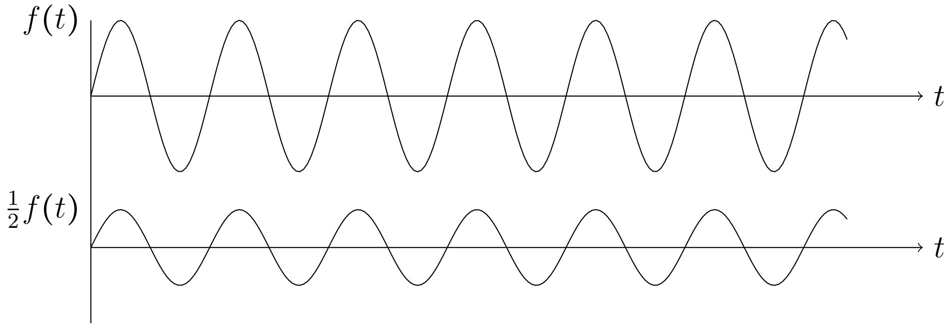


Figure 15: Scaling the amplitude by $a = \frac{1}{2}$

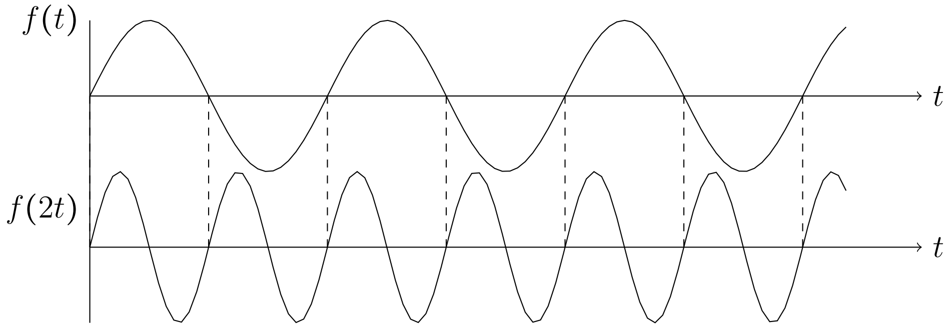


Figure 16: Time-scaling by a factor of 2

6.2 Time Shifting

Time-shifting, in principle, delays the signal or advances it.

Delaying a signal $f(t)$ by T_0 time units corresponds to the signal,

$$g(t) = f(t - T_0).$$

As an example, consider 3 cycles of a triangle wave [6] shown in Figure 17. Keep in mind that T_0 can be any real number, positive or negative. If T_0 is negative, the waveform will

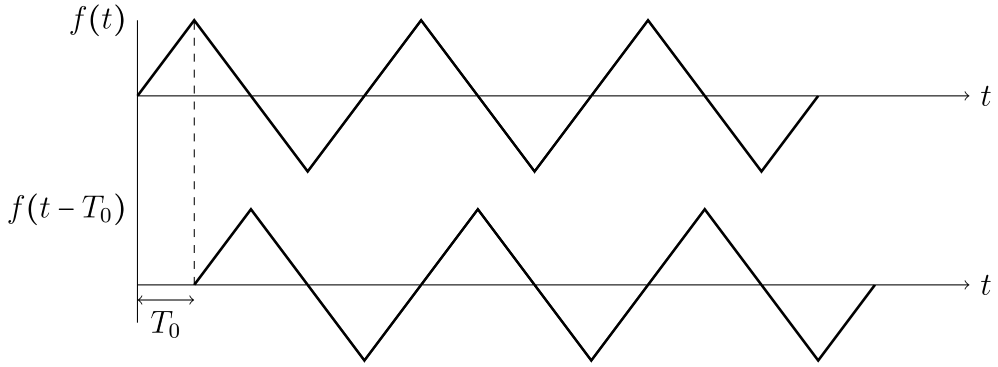


Figure 17: Time-shifting by T_0

[6] only 3 cycles are considered just for illustration, the same idea works for periodic waveforms too

be shifted to the left. It is evident that if $g(t) = f(t - T_0)$, then $f(t) = g(t + T_0)$, which corresponds to $g(\cdot)$ being shifted to the right along the time-axis.

7 Echo-System

This section summarizes some of the properties above with an example. We will describe a system with echos, which will involve scaling and time-shifting of input signals. In particular, an echo is a time delayed version of the original signal. We will introduce the notion of adding signals, wherein the original signal and echoes are added to obtain a new waveform. A practical example of a system with echoes happens in mobile cellular communications. The echo phenomenon there is known as **multi-path propagation**. Here a signal transmitted from the antenna reaches the destination antenna through multiple-paths with different delays, see figure. Apart from its relevance to a key technology sector, the above propagation model is of fundamental importance. It is a prototype of many systems that we will learn in this course.

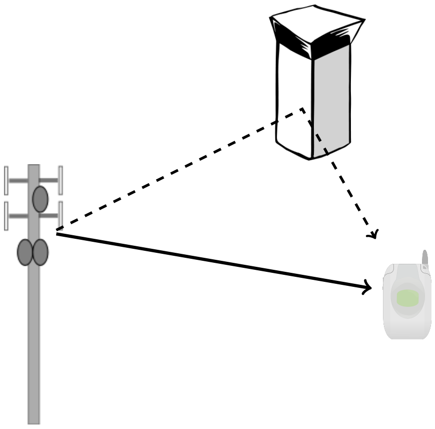


Figure 18: Multi-path propagation in mobile communication

Example 5. *The expression,*

$$y[n] = \frac{1}{2}x[n] + \frac{1}{2}x[n-1] \tag{18}$$

defines a system from the class of discrete-time signals to itself.

Exercise 7. *Justify the usage 'a low-pass filter' for the system in* (18).

Exercise 8. *Can you suggest one change which will render the system to a high-pass one.*

The above examples are that of linear systems. There are non-linear systems too. In fact, most systems in real life has some non-linearities (we all have it), and we will first learn how to live with linear approximations. We can *launch* ourselves to bigger '*depths*' once we master linear systems.

Let us now have a more formal definition of linear systems.

8 Linear Systems

Let us go back to the previous example . This belongs to the class of 'linear' systems. In Euclidean coordinates *linear* is synonymous with the geometric *path* (hyper-plane) defined by

$$a_1x_1 + a_2x_2 + \cdot\cdot + a_kx_k = c$$

where x_i is the i^{th} coordinate in $\mathbb{R}^n$. We can extend this to abstract spaces, by treating every member(element) of the space as a coordinate. For example, linear combination of sinusoids, where every coordinate is a sine wave (how many parameters do you need to specify a wave?). While an Euclidean *line* and sum of sine-waves are straightforward to imagine, what connection does these have to linear systems?, or plainly, what is a linear system?. To answer this, we give a precise characterization.

Linear System
A system defined by the mapping $y = f(x)$ is linear iff the following two properties hold:
(i)Superposition :
If $y_1 = f(x_1)$ and $y_2 = f(x_2)$ are the outputs produced by inputs x_1 and x_2 respectively, then

$$y_1 + y_2 = f(x_1) + f(x_2) = f(x_1 + x_2) \tag{19}$$

(ii)Homogeneity :
If $y_1 = f(x_1)$ and $x_2 = ax_1$, $a \in \mathbb{R}$, then

$$y_2 = f(x_2) = af(x_1) \tag{20}$$

Exercise 9. *For a linear system defined as above, show that*

$$f(ax_1 + bx_2) = af(x_1) + bf(x_2), \forall a, b \in \mathbb{R} \tag{21}$$

Exercise 10. *Taking equation* (21) *as the definition of linearity, show that equations* (19) – (20) *holds. Hence, deduce that both definitions are equivalent.*

Most of the systems that we deal with are linear. But we will occasionally encounter some non-linear ones. A usually employed trick is to approximate the non-linear system by a collection of linear ones. Let us look at another simple example.

Example 6. *Imagine a coffee vending machine which accepts coins as input. Every rupee as input makes the machine deliver α ml of coffee, where α is some number. Notice that the input output relationship here is linear. .*

Exercise 11. *Write a proof that the relationship in the above example is linear.*

In India, most coffee machines have a dedicated coffee vendor (we do not know why, but let us find some use for this guy). Our coffee vendor starts accepting dollars now (a fashionable thing to do), but not his poor coffee machine. But exchange rates are flashed every minute, and using that he finds (based on his rule) the equivalent Indian coins, and delivers coffee. From the point of view of dollar drawing people, the system (unit price) is dynamically varying in time. In other words, to get a glass of coffee at a certain time of the day, she (or her lover who pays) should check the price at that time, or the conversion rate. This brings to us the notion of time-varying systems. Let us start with defining a time-invariant system.

9 Time Invariance

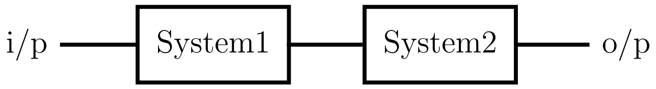
Time Invariance
Consider a well-defined system, where an input $x(t)$ corresponds to an output of $y(t)$, $\forall t \in \mathbb{R}$. The system is *time-invariant* iff,
$x(t-\tau)$ causes an output of $y(t-\tau)$, $\forall t, \tau \in \mathbb{R}$
Time Variance
A system is time varying if it is not *time-invariant* as in the above definition.

Exercise 12. *From the definition of time-invariance, our coffee example seems to be time-invariant when we pay rupees and time-variant when we buy in dollars. Is this a discrepancy. Explain your answer.*

Systems which are *linear* and *time invariant* belongs to an important class, and for most of the lectures we consider only such systems. We give it a pet name, **LTI** systems. The reader can now guess what **LTV** stands for.

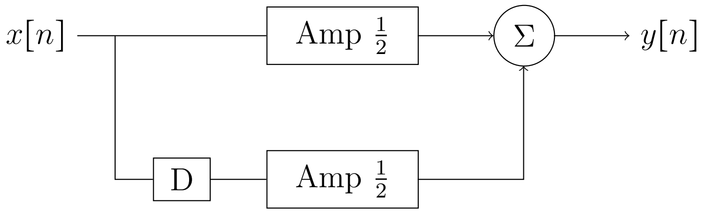
9.1 Cascade of Systems

We can connect systems in a sequential manner, by feeding the output of the first as input to the second. Some of you might have encountered this while studying *cascade amplifiers.* Assuming that such operations are well-defined, many complex systems can be considered as cascade of simple ones.



Exercise 13. *Show that a cascade of (i) LTI and LTV is LTV. What about an LTV and another LTV.*

We should not be prejudiced about having only serial connection of systems. Parallel systems not only are possible, but also are key to our understanding of complex systems. For illustration, let us express (18) in parallel form.



Here the box D is a delay element which delays the sample by one fundamental interval.

Exercise 14. *Comment whether LTI parallel LTI is still LTI. What about LTI parallel LTV?.*

We have seen that time-variations pose number of problems, increasing the complexity of analysis. What will we do as a coffee vendor (engineer) to make things simple. One idea is to take the closing exchange rate at a day, and use it as a fixed conversion rate for the whole of the next day, ignoring changes within the trading hours. So within a day, we have a linear time-invariant system. Come next day, we may have another LTI system, depending on what the rates are. Do this for every day, we have approximated the time-variance by a collection of time-invariant ones, making life easy for all. This is the reason why LTIs are important, not only in theory, but also in practice. A keen reader may have observed that, to reduce complexity further we can fix the exchange rate for weeks or months instead of a 24 hour basis.

Let us now zoom into the LTI class. Assume that the vendor had fixed the exchange rates for a long time (say an average butterfly lifetime). He now starts accepting all currencies, converting to rupees first and then business as usual. So even in the LTI class, converting the input into a form which is simple and easy to manipulate, is important. Those who attended the first lecture will realize that we were trying to motivate the very same thing.

Example 7. *A computer is like our coffee shop!. Yes, you are in the right class. It accepts a program in some language (currency), converts to machine language, executes and in turn outputs some service (coffee).*

Exercise 15. *What is the standard currency that the computer accepts. (give it a try, there is no right or wrong answer)*

Exercise 16. *Is the computer that we considered in previous example time-invariant. Comment on the linearity of the system.*

10 Time Variance

We already mentioned the naive fact that time variance means not time-invariant. To complete the story, let us have a closer look at time-variance. In several physical examples, time variance is caused by the relative movement of the signal source (input) and the system. A well known example is Doppler Spread, which corresponds to a frequency shift. (Here we describe a simplified version, which is just to convey the concept. Please see a physics textbook or wikipedia for a more systematic and correct explanation.) If the relative motion takes the source and observer closer, then a sine-wave, assuming no amplitude degradation over the path (and not considering path delays) will be observed as

$$sin(2\pi f_c t) \to \sin 2\pi(f_c + f_d)t \tag{22}$$

(Please note that the above is merely a mathematically convenient representation than the physics behind it. In particular, a sine wave starting at a certain time, has much more frequency components than the *visible* number of cycles per time unit.)

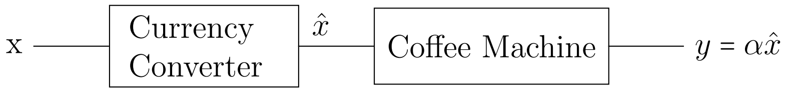
Exercise 17. *Argue that the above system is time-variant.*

This gives us a way to check linear time-invariance of an arbitrary system (given to us inside a black box with input and output ports). Feed a sinusoid as input and check whether there is any new frequency at the output. If yes, there is some time-varying element inside the black box.

Exercise 18. *A scientist declared that a sinusoid of frequency f_c produced a frequency of $2f_c$ at the output of a time-invariant system. Comment whether he is making a mistake and why?.*

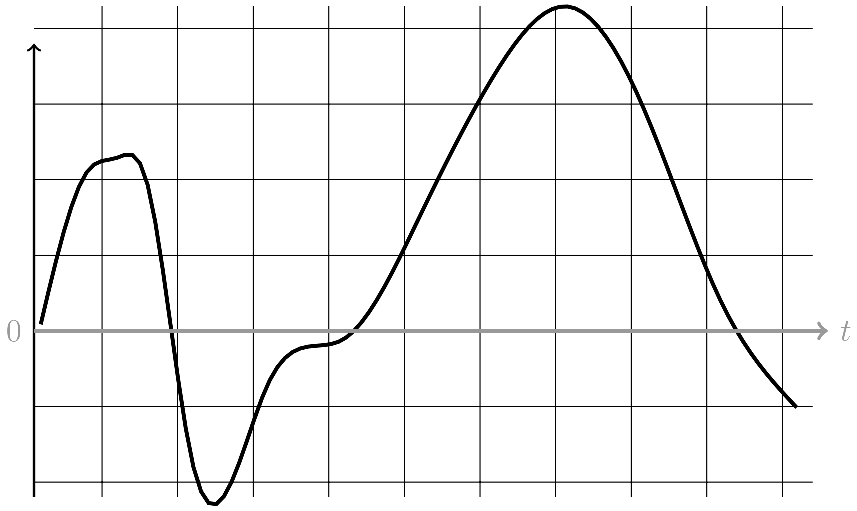
11 Currency Conversion on Signals

The example that we saw motivates us to think of defining a base currency for LTI systems to operate up on. Pictorially, the coffee machine example with time-invariant assumption can be represented as



In picture, $\hat{x}$ is the equivalent Indian currency which the machine can easily manipulate. Let us try to emulate the first step of converting x to $\hat{x}$, but now x is a signal given to us, instead of some currency.

Let $x(t) \in \mathcal{C}^0$, one such signal is showed below for some interval of time in the positive axis. It is assumed that the signal is continuous and well-defined on the rest of the axis too.



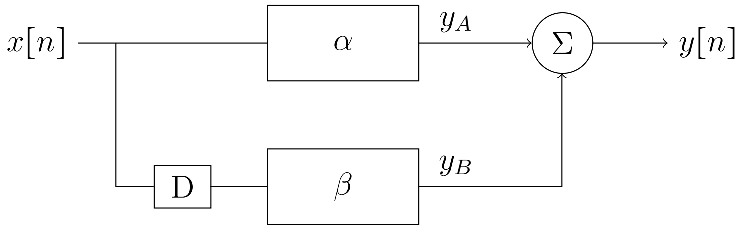
How will we represent this signal in a concise manner for the shown interval. At first, it *appears* that we need an infinite number values, say if we store $x(t)$ for every rational t. This thought leads to the further (simpler) idea that we can store the values at equi-distant intervals (at integer multples of a given time interval). Most students may rightly guess that if the samples are far apart, we may loose the 'fidelity' of the original signal, and on the other hand, too many samples will make life difficult. Is there a middle line, the Sampling Theorem deals with it, which we will cover soon. However, we will first focus on an alternate representation called Fourier Series for periodic signals. And guess what, studying Fourier Series will give us the Sampling Theorem with little extra effort. Our goal now is to explore the thread of events and connections in this treasure hunt.

12 Linear System in Discrete Time

Before jumping to the intricacies of Fourier series, let us re-examine linear systems. This will enable us to apply the Fourier Integrals to linear systems also. We argued that the linear time-invariant systems are such an important class of systems. One of the main examples that we considered was the operation of the form,

$$y[n] = \alpha x[n] + \beta x[n-1]. \tag{23}$$

We convinced ourselves that for real-valued signals defined on integer indices, this is a well defined system. In reality, we are being a bit partial to the signals i.e., given a piece of paper, you can write down the values of the signals, be it at the input or the output. What about systems?. Seems that we clearly like signals. To be fair, let us make a similar representation for systems. In Lecture-notes 1, the following parallel arrangement was used to represent the system in (23).



For simplicity, assume that $x[n] = 0, \forall n < 0$. Let us observe the waveforms at points A and B, when $n \geq 0$.

$$\begin{array}{rcl} (y_A[n])_{n\geq 0} & = & (\ \ \alpha x[0] \quad \alpha x[1] \quad \alpha x[2] \quad \cdot\cdot \quad \alpha x[k] \qquad \cdot\cdot\ ) \\ (y_B[n])_{n\geq 0} & = & (\qquad 0 \qquad \beta x[0] \quad \beta x[1] \quad \cdot\cdot \quad \beta x[k-1] \quad \cdot\cdot\ ) \end{array} \tag{24}$$

If we combine Y_A and Y_B in to a matrix form as,

$$Y_\Sigma = \begin{bmatrix} (y_A[n])_{n\geq 0} \\ (y_B[n])_{n\geq 0} \end{bmatrix}, \tag{25}$$

then the output at each n is nothing but the sum of the n^{th} column of Y_Σ. This gives us a handle on representing systems. We will simply denote the system as

$$\begin{bmatrix} \alpha & \beta \end{bmatrix} \tag{26}$$

More generally, a discrete system which at any time takes a linear combination of $M+1$ consecutive samples can be represented by a parallel arrangement of M delays. We will simply represent this system as

$$h = (h[n])_{n\geq 0} = (h[0] \quad h[1] \quad \cdot\cdot \quad h[M]) \tag{27}$$

Sometimes, when the context is clear, to have some economy of space, we will write $h[j]$ simply as h_j, excluding the square brackets, the same applies to $x[i]$, which is written as x_i. Can we write a relation similar to (25) for a general discrete-time system h. It is straight forward as shown in the matrix representation below.

$$\begin{vmatrix} h_0x_0 & h_0x_1 & h_0x_2 & \cdot\cdot & h_0x_M & h_0x_{M+1} & \cdot\cdot \\ 0 & h_1x_0 & h_1x_1 & \cdot\cdot & h_1x_{M-1} & h_1x_M & \cdot\cdot \\ \cdot & \cdot & \cdot & \cdot\cdot & \cdot & \cdot & \cdot\cdot \\ \cdot & \cdot & \cdot & \cdot\cdot & \cdot & \cdot & \cdot\cdot \\ 0 & 0 & 0 & \cdot\cdot & h_Mx_0 & h_Mx_1 & \cdot\cdot \end{vmatrix}$$

Observe the sum of each column corresponds to the output of the linear time invariant system defined by $h = (h_0\, h_1 \cdot\cdot\, h_M)$. We can represent this concisely as

$$y[n] = \sum_k h[k]x[n-k]. \tag{28}$$

The sum represented above is known as **convolution**. We will often write this as,

$$y[n] = h[n] * x[n], \tag{29}$$

where $*$ stands for the convolution sum. When the time index n is clear from the context, we may even shorten this as

$$y = h * x. \tag{30}$$

For those who like matrices, the convolution sum can be written in a concise pleasing form.

$$y = h\bar{X} \tag{31}$$

where $\bar{X}$ is a matrix with the input vector $(x[n])$ in first row, and k^{th} row as $(x[n])$ shifted to the right by $k-1$ elements, i.e. $(x[n-k+1])$. There will be $M+1$ such rows, that the multiplication is well-defined. When $M < \infty$, the system h is known as a finite impulse response filter (**FIR filter**).

There are infinite impulse response filters (**IIR**) too, but we will study it after learning how to design FIR ones. By the way, FIR filter design is all about choosing the coefficients $h_0 \cdot\cdot h_M$. We have to choose M as well as the values. Higher M, as expected, gives us more control on the filter, but will require more computations to obtain the convolution sum.

There is something more funny about the convolution sum. Do you remember seeing this operation illustrated in the matrix form before equation (28) anywhere else. Guess hard and long, $13-14$ years being such a long time. Where were you at that time, likely to be in your second standard class, where the teacher with stick taught you multiplication on the board. Here is one such procedure.

$$\begin{array}{ccccccccc}
 & & 1 & 3 & 4 & 3 & 1 & \times \\
 & & & & 4 & 2 & 3 & \\
\cline{3-8}
 & & 3 & 9 & 12 & 9 & 3 & \\
 & 2 & 6 & 8 & 6 & 2 & & \\
4 & 12 & 16 & 12 & 4 & & & \\
\hline
4 & 14 & 25 & 29 & 22 & 11 & 3 &
\end{array}$$

The final step is to move the carry from right to left, that we get a decimal representation. But the brave may ask, what if we don't use a decimal system. For example, the hexadecimals use A,B,C,D,E,F for numbers 10 to 15. Similarly a system based on $0, 1, \cdot\cdot 9$ and $A-Z$ can represent 35 unique numbers. In such a system, we will not have carry bits from left to right in the above multiplication, as all numbers are below 35.

Exercise 19. *Represent the output of our multiplication in a 35-mal number system as described above.*

This seems like an unnecessary divergence. The point we wish to make is that, multiplication without carry is conceivable and quite natural if we do not stick to just the decimals. With this, we present our main result.

Theorem 1. *Convolution has one to one correspondence to multiplication without carry addition.*

Exercise 20. *Prove the above theorem.*

The above theorem gives us a simple means to compute the convolution of any two discrete sequences. Typically, one will be the signal and other the system. Nevertheless, the question remains about carry. Why is the carry adding step omitted. To understand that, let us compare the multiplication example and convolution. Take the first number as a sequence of decimals, and imagine it as a signal. So the signal is $x[0] = 1, x[1] = 3, x[2] = 4, x[3] = 3$ and $x[4] = 1$. Similarly treat the second number as the system with $h[0] = 4, h[1] = 2$ and $h[2] = 3$. The system outputs $h_0 x_0$ at time 0, assuming that $x[n] = 0, \forall n < 0$. Let us consider two cases. If there was no carry to the final digit in our multiplication, the system output is same as the last digit in multiplication. On the other hand, if there was a carry in our multiplication, we will know about it only in the next time instant, when the second column is summed up. But in the second instant, since we have already read out the sum of the first column as output, we cannot now add the carry to it. This is the reason that a carry addition step is missing in convolution.

13 Convolution in Continuous Time

Summations in discrete times are typically replaced by integrals in continuous time. One can actually guess the following formula from Equation (28).

$$y(t) = \int_{\tau} h(\tau) x(t-\tau) d\tau \tag{32}$$

This extension is possible under the assumption that the above Riemann integral is well defined. This is not the case with number of pathological inputs and systems. To develop the integral as above in a more concrete fashion, we need tools from measure theory, which are beyond the scope of this lectures. But this does not stop us from trying to understand why the above definition works at instants where the integral is well-defined. The important formalism that we need to carry from our understanding of the discrete case is the so called Kronecker delta or discrete impulse. It is defined as,

$$\delta[n-m] = \begin{cases} 1\,, & \text{whenever } n = m \\ 0\,, & \text{otherwise.} \end{cases} \tag{33}$$

In particular, with the definition above, we have

$$x[n] = x[n] * \delta[n]. \tag{34}$$

From this observation, we go ahead and define a continuous time impulse or a **Dirac measure** $\delta(t)$ as that function where for every $t \in \mathbb{R}$, the property

$$x(t) = x(t) * \delta(t), \tag{35}$$

holds for all signals $x(\cdot)$ for which the convolution integral is defined at t . The convolution symbol used above is for the integral as in (32).

$$\int_{\tau} x(\tau)\delta(t-\tau)d\tau = x(t) \tag{36}$$

With this definition notice that,

$$x(t) * \delta(t-s) = x(t-s). \tag{37}$$

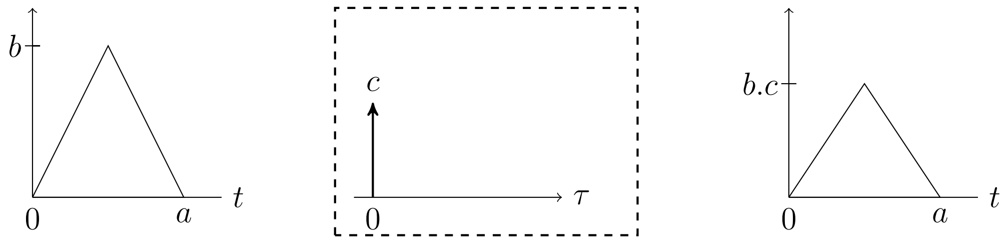
Exercise 21. *Prove the above assertion.*

Many engineering textbooks suggest to visualize an impulse as a rectangle with base tending to zero and the area kept unity as we shrink the base. i.e. the height is increased in a proportional fashion. While this is not such a bad idea in conveying that the imaginative sketch should go out of paper, it presents a difficulty in many situations where actual integrals are calculated. So once the concept is understood, we should always use the formalism in (36). With this representation, we can relate the Kronecker delta for the discrete-time and the Dirac measure in continuous time, and even use the same sketch to represent them. While relating this way, it is understood that a $\delta(t-s)$ in the system merely outputs the delayed signal, $x(t-s) \forall t, s \in \mathbb{R}$, analogous to a $\delta[n-k]$ in the system causing an output of $x[n-k]$ (remember our holiday party and payment in cheques).

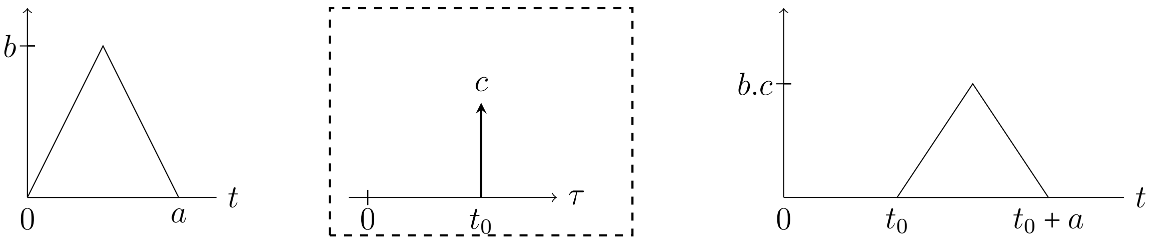
Exercise 22. *Evaluate*

$$\int_{t} x(t)\delta(t-s)dt. \tag{38}$$

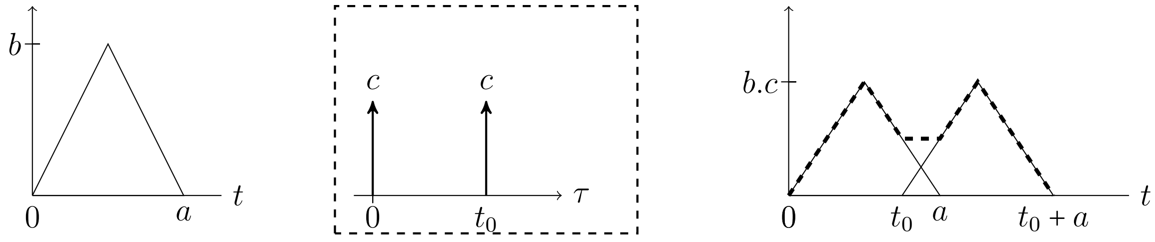
Example 8. *The following example illustrates convolution of a continuous time signal with an impulse (Dirac) at time zero. The impulse is having a weight c. i.e., its area is c instead of unity. (In the notes, we will always draw the impulse by an upward arrow with closed tip reaching up to c, but this may not be the standard convention in your disciplines.)*



Example 9. *What will happen if the impulse in previous example is shifted to time t_0.*



Example 10. *Suppose that instead of one impulse, we have two, one at time* 0 *and other at time t_0. This system is linear(why?). The system is the sum of the those in the last two examples. Since the input remains the same, the corresponding output will be the sum of the two outputs in the last examples.*



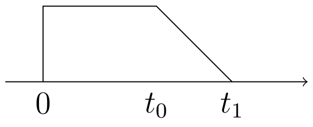
Observe that the big triangle regions at the output corresponds to the past two examples and the dashed region is the sum of those triangles, obtained by point-wise addition at each instant of time.

Exercise 23. *Consider a rectangle signal $r(t)$ defined as*

$$r(t) = \begin{cases} b \text{ if } 0 < t < a \\ 0 \text{ otherwise} \end{cases} . \tag{39}$$

What is $r(t) * r(t)$.

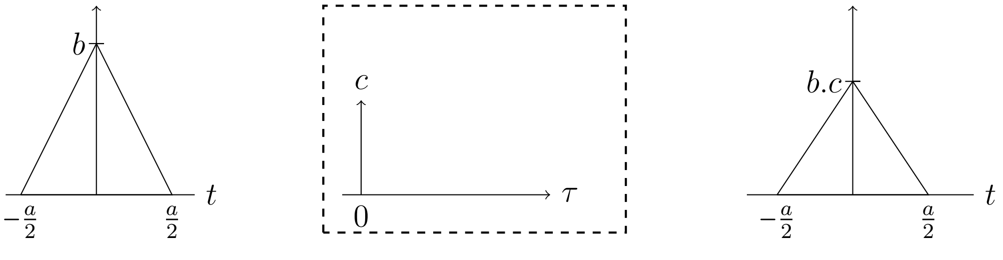
Exercise 24. *Consider applying the input shown below to the system in Example 10. Let $c = 1$ (unit impulse) and the height of the input waveform is also 1. Draw the output signal $\forall t$ when $t_0 = 2$ and $t_1 = 3$, with the time axis properly marked.*



14 Non-causal Systems

The last three examples considered causal systems. But convolution itself has no such restrictions, it works as well for non-causal systems too. The physical interpretations of causality apart, including non-causal systems in our study gives us more flexibility and adds an inherent symmetry in time to the system.

Example 11. *Consider the following signal*



Notice that though the system starts at time zero, there is non-zero output from time $-\frac{a}{2}$. This is in some sense analogous to people making each payment for our holiday party a few days in advance. We saw that delays caused a shift to the right. Advances do the reverse, a shift to the left. This system actually is causal, as the input values at any time t depends only on current and past values.

Now imagine that we interchange the system and signal in the previous example. Since the convolution formula remains the same, the output is the same as that in the example. But the system now becomes non-causal, as the output at time $t = -\frac{a}{2}_+$ depends on the input at time zero.

From now onward, we will not demarcate between causal and non-causal systems. It is not just for mere convenience, but a lot of notions that we are going to introduce needs both sides of the time axis. For example, a sine wave having just a single frequency has to span itself over all time, from $-\infty$ to ∞. This is not just an imaginary concept. Consider yourself walking inside a circular tube, let us say anticlockwise is the positive direction, and clockwise negative. It is a never ending journey, and we can extend this to time too.

15 Time Reversion

We have used the convention all along that time t runs from left to right. Imagine a traveling wave (say a wave in the ocean), and you are watching it from a podium which points south, and waves are traveling the east to west direction. The wave as such moves on the surface of the sea. So the values close to time zero (creation of the wave) are actually moving to the west (right) in space and more and more values added to the tail part. Similarly, a signal can be visualized as being passed through the system (just like our tourists to Goa). In this case, notice that the time is reversed. The first person to undertake the trip comes out first and keep walking forever through the calendar.

This time-reversal is actually inherent in the multiplication without carry that we described. So another way of computing the convolution sum is to take the signal, reverse it in time, and slide it through the system. In every-time instant compute the cumulative effect, i.e., each signal value which is inside the system is weighted by the corresponding segment of the system and all such effects are added up. But this is not anything new. We actually started with this idea in the first section of this chapter. We repeat it here to facilitate computing convolutions in continuous time.

$$y(t) = \int_{\tau} x(t-\tau)h(\tau)d\tau \tag{40}$$

Notice that there is a reciprocity between signals and systems. The output will be unchanged if you interchange a signal and the system. This helps in our computation. Instead of time-reversing the signal, we can keep the signal and time-reverse the system, and the output will be the same. In many cases that we consider, signal sequences are longer than the system (filter), so it is better to reverse the system sequence.

$$y(t) = \int_{\tau} h(t-\tau)x(\tau)d\tau \tag{41}$$

On careful observation, the above equation reveals that convolution can be determined in the following algorithmic fashion.

Exercise 25. *Consider a rectangle signal $r(t)$ defined as*

$$r(t) = \begin{cases} b \ \mathit{if}\, 0 < t < a \\ 0 \ \mathit{otherwise} \end{cases} . \tag{42}$$

What is $r(t) * r(t)$.

1. Obtain the time reversed system $h(-\tau)$. (This implies reflecting (as a mirror image) the system values around the origin, i.e. $\tau = 0$.)

2. For every time instant t,

   (a) we shift $h(-\tau)$ to the right by t units to get $h(-(\tau - t)) = h(t - \tau)$.

   (b) Initialize $y(t) = 0$;

   (c) Find the sum over τ (see note [a])

   $$y(t) = \int_{\tau} x(\tau)h(t - \tau)d\tau. \tag{43}$$

   (d) Output $y(t)$.

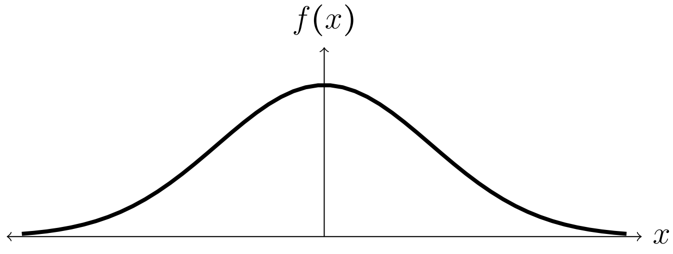
3. End if no more $y(t)$ is required.

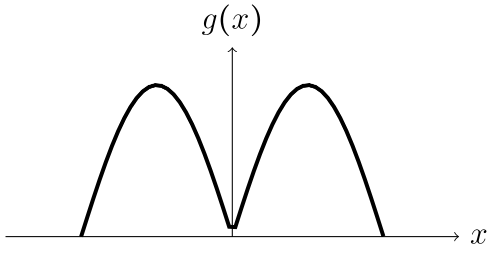
[a] The Riemann integral in equation (43) can be written as a summation in algorithmic implementation, by partitioning τ. Step (b) is to facilitate this recursive computation

Exercise 26. *Did you use the above technique while doing Exercise 24. If not, try it this way and compare both the answers.*

16 A Computational Simplification

Suppose you are given two continuous functions $f(x)$ and $g(x)$, say with bounded support. How can we program a computer to computer $f(x) * g(x)$. This may appear a dubious question at first, the computer is a fixed precision device and how are we going to compute integrals using this. Notice, however, that this problem is more fundamental than convolution, and numerical integration is a powerful tool in such contexts. So can we find approximate numerical values to the output of convolution using numerical techniques? This illustrated by an example below, for $f(x) = \exp(-x^2), x \in [-2, 2]$ and $g(x) = |sin(\pi x)|, x \in [-1, 1]$.





Observe that $f(x)$ can be approximated by a piece-wise constant function. A piece-wise constant function is nothing but an interpolation of equi-spaced samples of $f(x)$. We already learnt at the start of the chapter that piecewise constant interpolation can be written as a convolution of the input samples with a rectangular waveform. This is illustrated in Figure 19. Thus $f[x] \approx f_n * h(x)$, where $h(x)$ is a rectangle of unit height and width τ, and $f_n = f(n\tau), n \in \mathbb{Z}$. Here τ is the time interval between consecutive input samples. In a similar fashion, we can write $g(x)$ also as a convolution, say $g(x) \approx g_n * h(x)$, where $g_n = g(n\tau), n \in \mathbb{Z}$. This is shown in Figure 20.

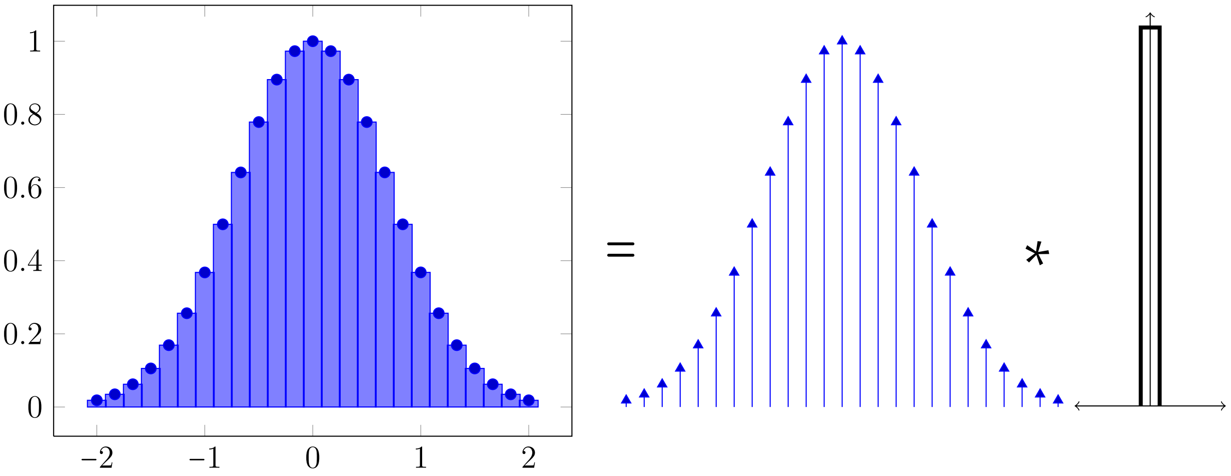


Figure 19: Piecewise constant approximation to $f(x)$

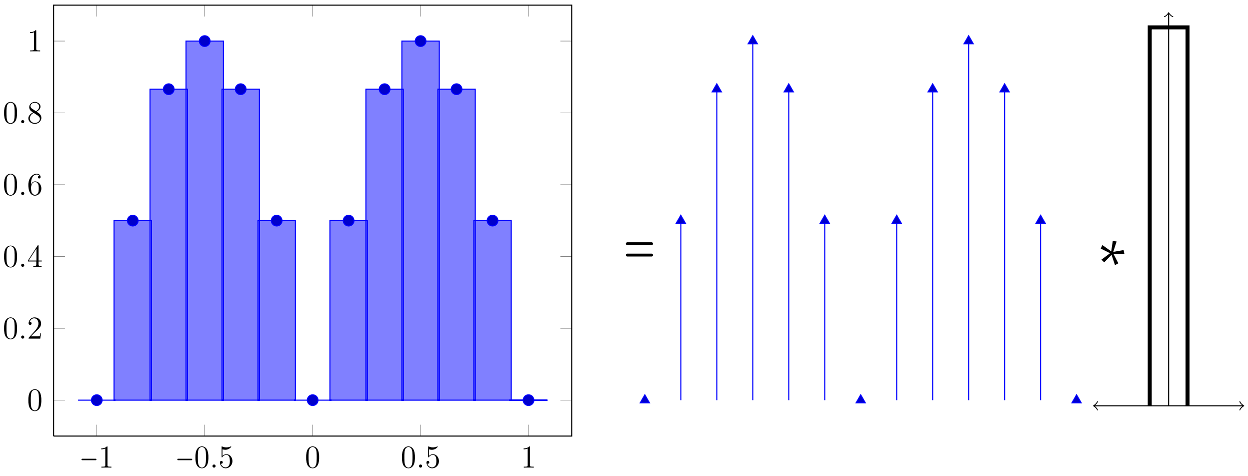


Figure 20: Piecewise constant approximation to $g(x)$

Clearly

$$
\begin{aligned}
f(x) * g(x) &\approx f_n * h(x) * g_n * h(x) && (44) \\
&= f_n * g_n * h(x) * h(x) && (45) \\
&= f_n * g_n * \Delta(x), && (46)
\end{aligned}
$$

where $\Delta(x)$ is a triangle wave-form centered at origin with width 2τ and height τ. In other words, we can find the discrete convolution $f_n * g_n$ first and then linearly interpolate the resulting values to obtain an approximation to $f(x) * g(x)$. This provides a algorithmic framework to evaluate the convolution of most functions that we encounter. For integrable functions, the approximation gives the actual values in the limit of $\tau \to 0$.

We like to remind the reader that both the functions $f(x)$ and $g(x)$ should be sampled at the same spacing τ for the above procedure to work. To exemplify this, we have drawn the approximations of Figures 19 and 20 in different scales of x−axis. In particular, $g(x)$ is only half as wide as $f(x)$, but they are drawn to the same display dimensions. The decompositions on the right side inherits the dimensions from the left most curve.