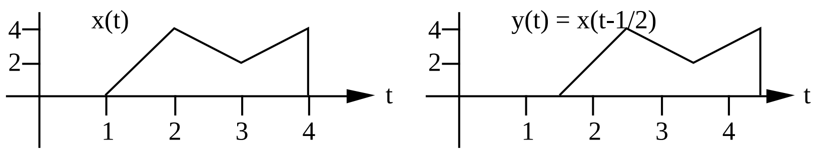
[9]**Time shifting:** If x(t) is a signal and T is some number, then the signal

$$y(t) = x(t-T)$$

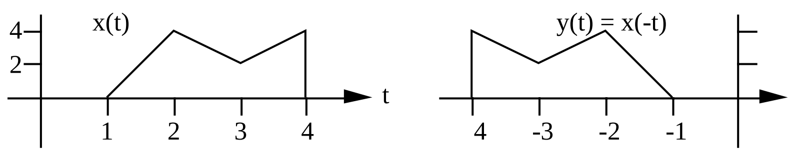
is a *time-shifted* version of x(t). That is, the value of y at time t is precisely the value of x at time t-T. This means that if T > 0, then as illustrated below, anything that "happens" in the signal x also happens in the signal y, but it happens T time units later in y than in x. Similarly, if T < 0, it happens T time units earlier in y. It is useful to remember the rule that a positive value of T leads to a right shift of the plot of x(t) and a negative value of T leads to a left shift.



Time reflection/reversal: The time reflected or time reversed version of a signal x(t) is

$$y(t) = x(-t).$$

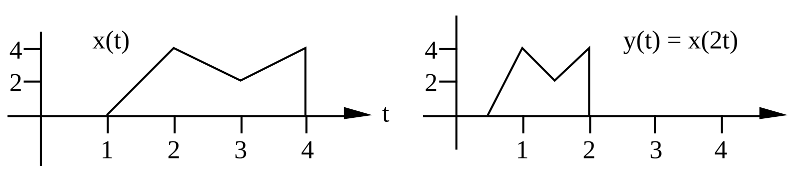
That is, whatever happens in x also happens in y, but at the negative of the time it happens in x.



Time scaling: The operation of *time-scaling* a signal x(t) produces a signal

$$y(t) = x(ct)$$

where c is some positive constant. If c > 1, this has the effect of "speeding up time" in the sense that the value of y at time t is the value of x at time ct, which is a later time. Alternatively, whatever happens in x in the time interval $[t_1,t_2]$ now happens in y in the earlier and shorter time interval $[t_1/c, t_2/c]$.



This is the one property that for which the discrete-time case includes an extra wrinkle. Specifically, in discrete-time, the time values must be an integer. Therefore, if we take

$$y[n] = x[cn],$$

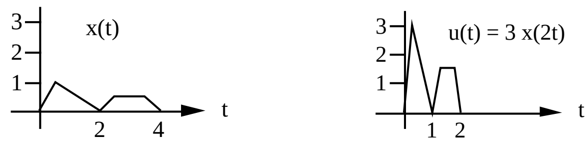
then c needs to be an integer.

Combinations of the above operations: In the future we will frequently encounter signals obtained by combining several of the operations introduced above, for example,

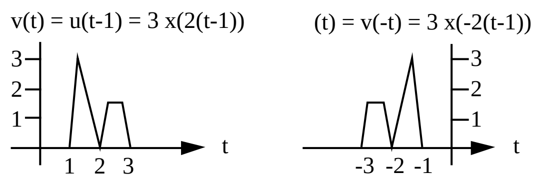
$$y(t) = 3\ x(-2(t-1))\ .$$

[9]this page 17 supercedes the previous page 17. The changes are to the last paragraph.

To figure out what signal this is, it is useful to introduce some intermediate signals. For example, in the above, we might start by plotting x(t) and u(t) = 3 x(2t), as shown next.



which is an amplitude scaling and time scaling of x(t). Next, one might plot v(t) = u(t-1) = 3 x(2(t-1)), which is a time shifting of u(t) by one time unit. Finally, we plot y(t) = v(-t) = 3 x(-2(t-1)) .



Note that you can also find y(t) by applying the scaling, time shifting and time reversal in some order, or by applying several at a time. But until you are very experienced, it is advisable to apply only one or two at a time.

B. Elementary Operations on Two or More Signals

Summing: As its name suggests, this is simply the operation of creating a new signal as the sum of two or more signals, as in

$$z(t) = x(t) + y(t) .$$

More specifically, the value of z at time each time t is the sum of x at time t and y at time t.

Linear combining: Linear combining is like summing except that we allow amplitude scaling (i.e. multiply the signals by constants) in addition to summing, as in

$$y(t) = 3\, x_1(t) + 4\, x_2(t) - 2\, x_3(t) .$$

In this case, y(t) is said to be a *linear combination* of $x_1(t)$, $x_2(t)$ and $x_3(t)$. The scale factors multiplying the x(t)'s are often called *coefficients*.

Linear combinations arise in a several ways. As one example, sometimes we are given a collection of signals, say $x_1(t)$, $x_2(t)$ and $x_3(t)$ and are asked to *synthesize* another signal y(t) as a linear combination of the signals in the collection. For example, suppose we need to create the signal y(t), but our hardware can only of produce signals $x_1(t)$, $x_2(t)$ and $x_3(t)$ and perform linear combinations. Often, it is not possible to exactly synthesize y(t) from the given collection and the synthesis must necessarily be approximate.

As another example, sometimes we are given a signal z(t) that is known to be a linear combination of $x_1(t)$, $x_2(t)$ and $x_3(t)$, and we are asked to find the scale factors. This task, which is called *analysis*, happens for example in communications systems, where the scale factors determine the information carried by the signal y(t). It also happens in *Fourier analysis*, to be discussed considerably throughout the course, where we consider a signal y(t) to be the linear combination of sinusoidal signals with different fundamental frequencies. Later in the course, we will discuss systematic procedures for finding the coefficients that make a linear combination of a given collection of signals $x_1(t)$, $x_2(t)$, ..., $x_n(t)$ match a given signal y(t), exactly or as well as possible. Such analysis techniques are also needed for the synthesis task discussed in the previous paragraph.

Multiplying: As its name suggest, this is simply the operation of creating a new signal as the product of two or more signals, as in

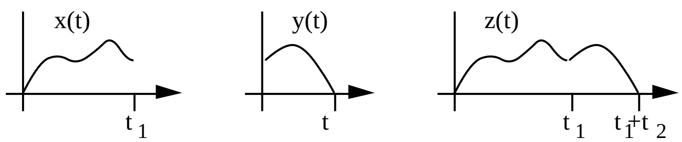
$$z(t) = x(t)\, y(t) .$$

More specifically, the value of z at time each time t is the product of x at time t and y at time t.

Signal multiplication is a basic operation of most radio transmitters which, as in the example of AM radio described earlier, typically multiply a sinusoidal signal by some information bearing signal.

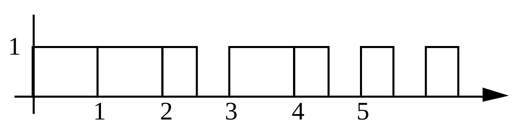
Concatenating: *Concatenation* is the process of adding one signal to the end of another. For example if $x(t)$ is a signal with support interval $(0,t_1)$ and $y(t)$ is a signal with support $(0,t_2)$, then as illustrated below their concatenation is the signal

$$z(t) = \begin{cases} x(t), & t \le t_1 \\ y(t-t_1), & t > t_1 \end{cases}$$



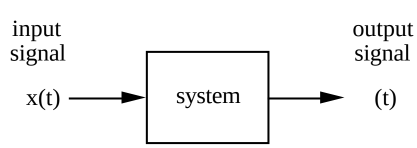
Concatenation happens, for example, in digital communications where, for example, to transmit a sequence at the rate of one bit every T seconds, there is a signal $s_0(t)$ with support $(0,T)$ used to send 0's, a signal $s_1(t)$ also with support $(0,T)$ used to send 1's, and the transmitted signal is the concatenation of these. For example, when the signals shown below

are used to send the binary sequence 0,0,1,0,1,1,1,... the transmitted signal[10] is



Concluding Remarks

The signal operations discussed in this section are elementary operations that are used in a variety of situations. One may view them as basic tools or building blocks. The signal operations described in the next section are more sophisticated operations, which are developed with some specific task in mind. They can be thought of as *systems* that is, when the operation is applied to a signal $x(t)$, the signal $x(t)$ is viewed as the input to a *system* that performs the operation and produces at its output another signal $y(t)$, which is the result of the operation. In such cases, we often draw a block diagram like the one shown below. Much of the course will be devoted to designing systems to perform the tasks described in the next section.



[10] As usual, vertical lines are shown just emphasize the transitions between transmitted bits, as well as the jumps from 0 to 1 and 1 to 0.

Revised Outline:

Introduction to Signals

I. Elementary Signal Concepts
   A. Signal Definition and Signal Descriptions
   B. Elementary Signal Characteristics
      i Signal Support Characteristics
      ii. Signal Value Characteristics
      iii. Signal Shape Characteristics
   C. Two-Dimensional Signals

II. Elementary Signal Operations
   A. Elementary Operations on One Signal.
   B. Elementary Operations on Two or More Signals

III. Basic Signal Processing Tasks
   A. Signal Recovery/Extraction/Enhancement
   B. Signal Detection/Classification/Recognition
   C. Other Signal Processing Tasks

IV. Sinusoidal and Complex Exponential Signals
   This is discussed in Chapter 2 of the text, not in these notes.

III. Basic Signal Processing Tasks

In this section, we describe two broadly stated, nearly ubiquitous tasks that require the processing of signals. That is, there is need to develop systems that perform these tasks on signals. Much of the remainder of the course will be devoted to developing techniques to improve such systems.

The tasks have a similar flavor. In each, the signal to be processed contains a component that interests us and a component that does not. That is, the signal $r(t)$ to be processed can be modeled as

$$r(t) = s(t) + n(t),$$

where $r(t)$ is the component that interests us and $n(t)$ is the component that does not. For example, the component that interests us might be the signal produced by someone speaking into a microphone, and the component that does not might be the signal produced by background noise. In the first task, called variously *signal recovery* or *signal extraction* or *signal enhancement* or *noise reduction*, the goal is to recover the signal component $s(t)$ that interests us. For example, we might wish to recover the speech signal without the background noise. In the second task, called *signal detection* or *signal classification* or *signal recognition*, we wish to make a decision about the signal component that interests us. For example, we might wish to decide the identity of the speaker or what the speaker has said. These two tasks will be introduced in the next two subsections.

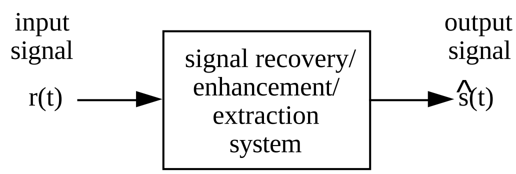
In each of the tasks, the noise $n(t)$ is not a known signal. If it were known, we could simply subtract it from $r(t)$, and there would be no need for a signal recovery or signal detection system. We also assume that the desired signal $s(t)$, or some aspect of it, is not known. If $s(t)$ were entirely known, we could dispense with $r(t)$, and simply display the signal $s(t)$. On the other hand, there must be something we do know about $s(t)$ and $n(t)$, such as some of their signal value or signal shape characteristics. Indeed, there must be something we know that is different for s(t) than for n(t). Otherwise, we will have no way to separate one from the other. For example, much of the course will be devoted to developing systems that work when s(t) and n(t) have spectra that differ in known ways, e.g. one contains only low frequencies and the other contains only high frequencies.

A. Signal Recovery/Extraction/Enhancement

Suppose we are given a signal r(t) with two components,

$$r(t) \;=\; s(t) + n(t)\ ,$$

and our task is to design a system, such as illustrated below, which processes r(t) in order to produce s(t), or more precisely an approximation $\hat{s}(t)$ to s(t).

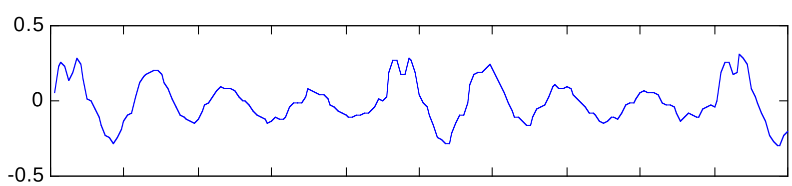


We consider r(t) to be the *original* or *measured* or *received* signal, s(t) to be the *desired signal*, and n(t) to be *noise*. This is sometimes called *signal enhancement* because we are enhancing the signal r(t). It is sometimes called *signal recovery* or *signal extraction*, because we are extracting or recovering the signal s(t) from the noise corrupted signal r(t). It is also called *noise reduction* or *noise suppression*, because it attempts to do precisely this.

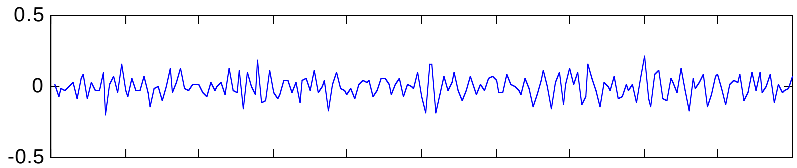
Examples of signals requiring recovery/extraction/enhancement include:

- An audio signal, especially when it is particularly faint, or when the microphone is part of a hearing aid, or when there is much background noise, such as in an automobile or helicopter or crowded cocktail party.
- A photograph or movie or video taken in faint light
- A signal being played back on an analog tape player (video or audio). Magnetic tapes introduce significant amounts of noise due to the granularity of the magnetic media.
- An AM or FM radio signal, or an analog TV signal, as it emerges from the receiving antenna. There is always lots of background noise, much of it due to other radio signals.
- A digital communication signal as it emerges from the receiving wire, antenna or other sensor. This signal must be extracted from background noise and from all other communication signals on the same medium.

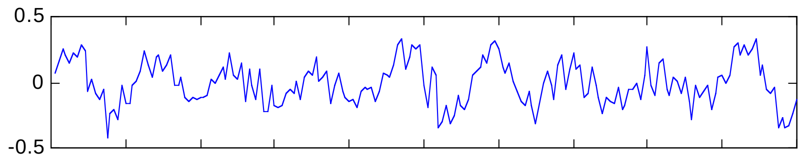
Linear Filters: There are many possible approaches to signal recovery. In this course we focus mostly on the approach called *linear filtering*, which is the most common approach. Let us introduce it with an example. Suppose s(t) is an audio signal, for example the one shown below.



and suppose r(t) = s(t) + n(t), where n(t) looks for example like the signal below,



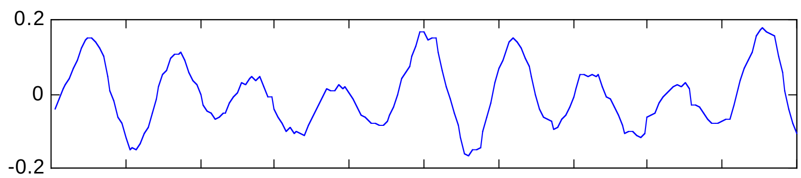
and r(t) is



Since the noise signal fluctuates more rapidly than the audio signal[11], a natural approach to reducing the noise is to use a *running-average filter*. That is, we design a system that replaces r(t) by an average of r(t) over an interval up to time t. Specifically, take the average over the interval (t-T,t), where T is chosen small enough that the audio signal s(t) changes little in such an interval and T is chosen large enough that the noise signal fluctuates many times in such an interval and, usually, averages to a small value. In other words, the running-average filter produces the output signal

$$\hat{s}(t) = \frac{1}{T} \int_{t-T}^{t} r(t')\ dt' \ .$$

When such a filter is applied to r(t), it has the effect of smoothing the r(t). In our example, it produces the signal shown below, which sounds much more like s(t) than r(t). Notice that the filtering has not only reduced the noise, but it has also modified the desired signal somewhat.



While the running average filter is fairly common, there are many other linear filters. As a precursor to introducing the range of possible linear filters, let us note that by applying the change of variables t'' = t'-t to the above integral, we may rewrite the running average filter as producing

$$\hat{s}(t) = \frac{1}{T} \int_{-T}^{0} r(t+t'')\ dt'' \ ,$$

which in turn may be rewritten as

$$\hat{s}(t) = \int_{-\infty}^{\infty} r(t+t'')\ w(t'')\ dt'' \ .$$

where

$$w(t'') = \begin{cases} \frac{1}{T}, & -T \le t'' \le 0 \\ 0, & \text{else} \end{cases} \ .$$

Other linear filters are obtained by replacing the function w(t''), which we call a *weighting function*, by something else. That is, the output is produced by a running average,

[11]This is the signal-shape charactertistic that differentiates the s(t) from n(t) in this example.

except that the average is with respect to a weighting function $w(t'')$. We obtain different linear filters by making different choices of $w(t'')$. For example, if we use

$$w(t'') = \begin{cases} e^{3t''}, & t'' \le 0 \\ 0, & t'' > 0 \end{cases}$$

Then

$$\hat{s}(t) = \int_{-\infty}^{0} r(t+t'')\, e^{3t''}\, dt''$$

In this case, we see that $\hat{s}(t)$ is the average of all past values of $r(t)$. However, in computing the average, past values are multiplied by exponentially decreasing weights.

By careful choice of the weighting function $w(t'')$, one can develop filters that do a better job of extracting the signal from the noise than the running average filter illustrated above. Quite a different sort of weighting function is needed to perform the complex task of extracting a single radio signal from all those at other frequencies. As the course progresses, we will develop better and better techniques for designing filters for extracting signals or suppressing noise.

Actually, in this course, we will focus primarily on discrete-time linear filters for filtering discrete-time signals. (See Chapters 5-8 of our text.) Specifically, a discrete-time filter performs the analogous operation

$$\hat{s}[n] = \sum_{k=-\infty}^{\infty} r[n+k]\, w[k] \ ,$$

where the $w[k]$'s are a sequence of weights that distinguish one linear filter from another. For example, if $w[k] = 1/M$, $k = -M+1,...,0$, then we obtain a discrete-time running average filter, which produces

$$\hat{s}[n] = \frac{1}{M} \sum_{k=n-M+1}^{n} r[k] \ .$$

There are more general forms of filters which we discuss in Chapters 5-8 (the discrete-time versions).

Quality Measure: As engineers, wherever possible we wish to quantify the goodness of the systems that we build. Thus, we need a quantitative measure of goodness for a signal recovery system. In this course, from time to time, we will use *mean-squared error* (MSE) as our measure of goodness. Specifically, if the signal $s(t)$ has support interval (t_1,t_2), then

$$MSE = \frac{1}{t_2-t_1} \int_{t_1}^{t_2} (s(t)-\hat{s}(t))^2\, dt$$

When using this quality measure, the goal of design is to minimize it.

Other Signal Recovery Tasks: There are other situations where the desired signal and noise are not simply added. Rather $r(t)$ depends on the desired signal $s(t)$ in some more complicated way. For example, in AM radio transmission the audio signal we wish to recover is the envelope of the transmitted signal (minus a constant), and it is desired to recover this audio signal from the transmitted signal plus noise. In tomographic imaging (e.g. X-ray, MRI, PET, etc.), the desired signal is a two or three-dimensional image, which must be extracted from a complex set of measurements. The same is true of synthetic aperture radar. These are advanced topics that will not be pursued in this course or in these notes.