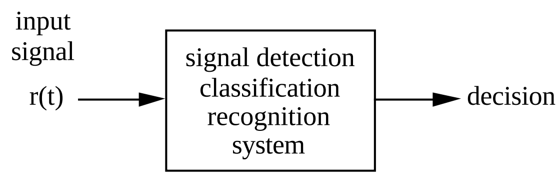
B. Signal Detection/Classification/Recognition

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) and produces a decision about s(t).



There are three closely related versions of this, introduced below along with examples.

1. **Signal/No Signal?** In this case, s(t) = 0 or s(t) = u(t), where u(t) is some known or partially known desired signal. From r(t) decide which of these two possibilities has occurred. This is considered to be a *detection* or *recognition* task because the goal is to *detect* or *recognize* whether or not u(t) has occurred. Specific examples:

   • Radar: Decide if the signal r(t) from the receive antenna contains a reflected pulse at time t_o. The same issues apply to sonar.

   • Dollar bill changer: Decide if the signal r(t) obtained by optically scanning a bill is due to a genuine dollar bill.

   • Fingerprint recognition: Decide if the signal r(t) obtained by optically scanning a fingerprint contains the fingerprint of John Smith. Similar tasks include recognition from retinal scans or voice prints.

   • Heart monitoring. Decide if an ekg signal r(t) contains a characteristic indicating a heart defect.

2. **Which Signal?** Here, $s(t) = v_1(t)$ or $v_2(t)$ or ... or $v_M(t)$, where M is some finite integer and the $v_i(t)$ are known signals. From r(t) decide which of the $v_i(t)$'s is contained in r(t). This is considered to be a *classification* or *recognition* task because the goal is to *classify* r(t) according to which $v_i(t)$ has occurred, or equivalently to *recognize* which $v_i(t)$ has occurred. Specific examples:

   • Digital communication receiver: Decide if the received signal r(t) contains the signal representing zero or the signal representing one. That is, the system must decide if the transmitter sent zero or one. In some systems, the transmitter has more than two signals that it might send, and so the receiver must make a multi-valued decision.

   • Optical character recognition: Decide if a character printed on paper is a or b or c or ... . This is especially challenging when the characters are handwritten.

   • Spoken word recognition: Decide what spoken word is present in the signal r(t) recorded by a microphone.

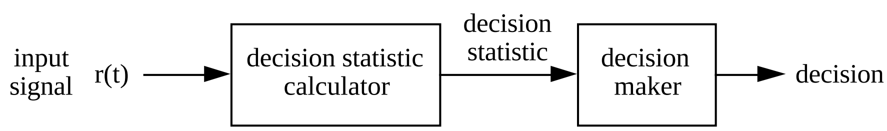
   • The "signal/no signal" task may be considered to be a special case of the "which signal task".

3. **Signal? And if So Which Signal?** This is a combination of the two previous subtasks -- s(t) = 0 or $v_1(t)$ or $v_2(t)$ or ... or $v_M(t)$. From r(t) decide whether or not s(t) = 0, and if not, decide which of the $v_i(t)$'s is contained in r(t). Examples:

- Digital communication receiver: Some digital communication systems operate *asynchronously* in the sense that the receiver does not know when the bits will be transmitted. In this case, the receiver must decide if a bit is present, and if so, is it a zero or a one.
- Personal identification system: Decide if a thumb has been placed on the electronic thumbpad, and if so, whose thumb.
- Touch-tone telephone decoder: Decide if the signal from a telephone contains a key press, and if so, which key has been pressed.
- Spoken word recognition: Decide is a word has been spoken and if so, what word.

For brevity, we will use the word *detection* as a broad term encompassing all of the above.

Detection Systems: As illustrated below, a detection system usually has two subsystems: the first processes the received signal in order to produce a number (or several numbers) from which a decision can be made. The second makes the decision based on the number (or numbers) produced by the first. The number or numbers produced by the first system are called *decision statistics* or *feature values,* and the first subsystem is called a *decision statistic calculator* or a *feature calculator*. The second subsystem is called the *decision maker* or *decision device*. We will consider two general types of detection systems, corresponding to two types of decision statistic generators -- *energy detectors* and *correlating detectors.*



Quality/Performance Measures: For detection systems, the most commonly used measure of performance is the *error frequency*, which as its name suggests, is simply the frequency with which its decisions are incorrect. We let the symbol f_e denote the error frequency. The typical goal is to design the detection system to minimize f_e.

In some situations, certain types of errors are more significant than others. For example, from the point of view of the owner of a dollar bill recognizer, classifying a counterfeit bill as valid is a more significant error than classifying a genuine dollar bill as invalid. In such cases, one will want to keep track of the frequency of the different types of errors. And one may choose to minimize the total frequency of errors subject to constraints on the frequencies of certain specific types of errors. For example, the owner of a dollar bill recognizer might insist that detector make as few errors as possible, subject to the constraint that it classifiy counterfeit bills as valid no more than one time in a million.

Energy Detectors for Deciding Signal/No Signal: For the "signal/no signal" task, the detector must decide whether $r(t)$ contains signal AND noise, i.e. $r(t) = u(t) + n(t)$, or just noise, i.e. $r(t) = n(t)$. Since it is natural to expect that $r(t)$ will have larger energy in the former case than in the latter, it is natural to choose the energy $E(r)$ of $r(t)$ as the decision statistic. (One would normally measure the energy of $r(t)$ over the support interval of $u(t)$.) The decision maker would then decide that $u(t)$ is present if the energy is sufficiently large, and would decide that $u(t)$ is not present otherwise. To make such a decision, one needs to specify a *threshold*, denoted τ, and the decision rule becomes

$$u(t) \text{ is present if } E(r) \geq \tau, \text{ and } u(t) \text{ is not present if } E(r) < \tau .$$

How to choose the threshold? The first thing to note that is that the noise signal $n(t)$ is usually random. That is, it is not known in advance, and it is different every time we measure it. In particular, the energy of the noise will vary from decision to decision. However, based on past experience, it is usually possible to estimate the average value of the noise energy, which we denote $\overline{E}(n)$. Then we can say that when $u(t)$ is not present, the signal $r(t) = n(t)$ has a random energy value, whose average is $\overline{E}(n)$. On the other hand, when the signal $u(t)$ is present, the energy of $r(t)$, though still random tends to be larger. Specifically, it ordinarily has average energy equal to $E(u) + \overline{E}(n)$. In summary, when the signal $u(t)$ is present, the average energy of $r(t)$ is $E(u) + \overline{E}(n)$, and when $u(t)$ is not present, the average energy of $r(t)$ is $\overline{E}(n)$. It is natural then to choose a threshold that lies half way between these two average energy values. That is, we choose

$$\tau = \frac{1}{2}\left(E(u) + \overline{E}(n)\right) + \frac{1}{2}\overline{E}(n) = \frac{1}{2}E(u) + \overline{E}(n) .$$

Energy detectors can also be used for the "which signal" task, provided the signals $v_1(t)$, $v_2(t)$, ... , $v_M(t)$ have sufficiently different energies -- so different that the differences will not be obscured by the noise. In this case, the typical decision maker strategy is to compare $E(r)$ to the average energies $E(v_1)+\overline{E}(n)$, $E(v_2)+\overline{E}(n)$, ..., $E(v_M)+\overline{E}(n)$ that one expects if the various $v_i(t)$'s were present. The decision maker then decides in favor of the signal $v_i(t)$ such that $E(v_i)+\overline{E}(n)$ is closest to $E(r)$.

Correlating Detectors for the "Which Signal Task": For the "which signal" task, an alternate and usually more effective method of detection (than energy detection) is to directly compare $r(t)$ to each of the signals $v_1(t)$, $v_2(t)$, ..., $v_M(t)$. Accordingly, we need a measure of similarity, and we will choose *correlation.* Specifically, the correlation between two continuous-time signals $x(t)$ and $y(t)$ is defined to be

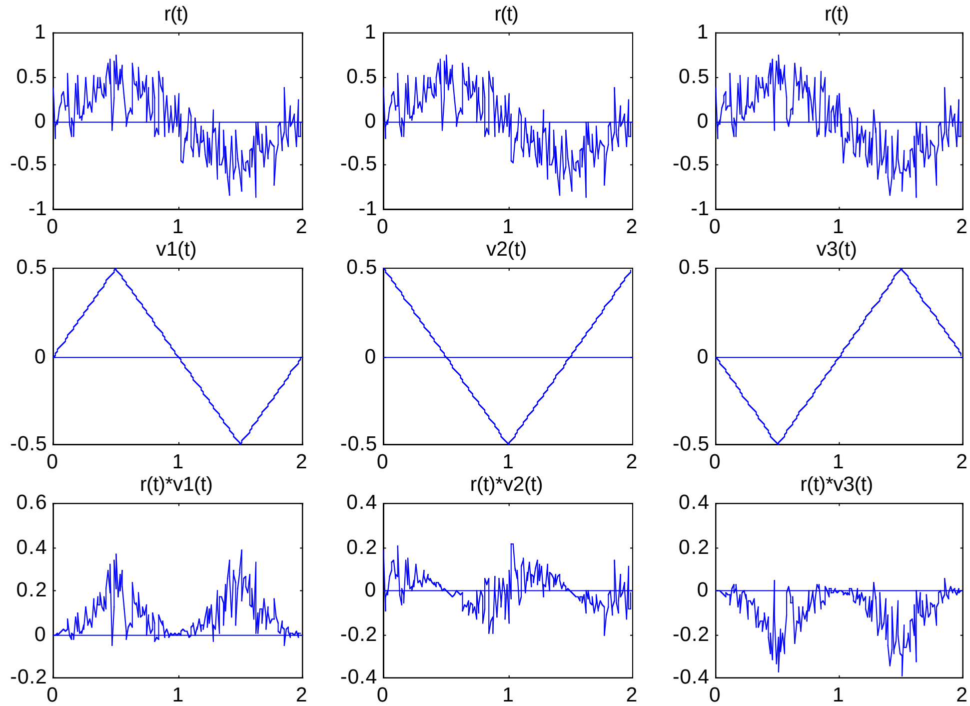
$$C(x,y) = \int_{t_1}^{t_2} x(t)\, y(t)\; dt ,$$

where (t_1,t_2) is the time interval of interest. Similarly, the correlation between two discrete-time signals $x[n]$ and $y[n]$ is defined to be

$$C(x,y) = \sum_{n_1}^{n_2} x[n]\, y[n] .$$

For brevity, we will continue the discussion presuming continuous-time signals. To see why correlation is a good measure of similarity to use in detection, consider the signal pairs shown below, in which a signal $r(t)$ is compared to the three possibilities $v_1(t)$, $v_2(t)$ and $v_3(t)$. To aid the comparisons, $r(t)$ is plotted above each signal. One can see that $r(t)$ and $v_1(t)$ are similar in that, roughly speaking, where one is positive, the other is as well; where one is negative the other is as well. Moreover, $r(t)$ roughly follows the shape of $v_1(t)$. On the other hand, the signals $r(t)$ and $v_2(t)$ are rather dissimilar. Where $v_2(t)$ is positive, $r(t)$ is sometimes negative; where $v_2(t)$ is increasing, $r(t)$ is sometimes decreasing. Finally, $r(t)$ and $v_3(t)$ are very dissimilar. Indeed, $r(t)$ is very much like the negative of $v_3(t)$. If one were to make a decision about which of the three signals $v_1(t)$, $v_2(t)$, $v_3(t)$ was contained in $r(t)$ based on visually comparing $r(t)$ to the these signals, one would clearly choose $v_1(t)$. And indeed this is correct, because $r(t)$ was generated by adding noise to $v_1(t)$.

Let's now consider how the same decision could be based on correlation. To do so, let's examine the value of correlation for each pair of signals. The product of each pair of signals is shown below the pair. Correlation is the integral of the product, i.e. the area under the plot of the product signal. For the first pair, the product is almost entirely positive, and the correlation is large. For the second pair, the product is approximately half negative and half positive, and the correlation is small because the positive and negative areas of the product tend to cancel each other. Finally, for the third pair, the product is mostly negative, and the correlation gives a large negative value.



$C(r,v_1) = 16.7$ $C(r,v_2) = 0.82$ $C(r,v_3) = -16.7$

If a detection system had to decide from the three correlation values which of the three signals $v_1(t)$, $v_2(t)$, $v_3(t)$ was contained in $r(t)$, clearly it should choose the one corresponding to the largest correlation, namely, $v_1(t)$.

Though correlation would work well in the example above, consider what would have happened if, for example, $v_2(t)$ were 100 times larger. In this case, it is easy to see that the correlation $C(r,v_2) = 82$, rather than 0.82. Thus even though v_2 has a very different shape than $r(t)$, a decision based solely on the size of the correlation would make the wrong decision. We can remedy this potential shortcoming by normalizing correlation. That is, it is better to make a decision based on *normalized correlation*, which is defined by

$$C_N(x,y) = \frac{C(x,y)}{\sqrt{E(x)}\sqrt{E(y)}} = \frac{1}{\sqrt{E(x)}\sqrt{E(y)}} \int_{t_1}^{t_2} x(t)\, y(t)\, dt$$

where $E(x)$ and $E(y)$ are the energies over the interval (t_1,t_2) of x and y, respectively. If the energies of the $v_i(t)$'s are the same, then signal $v_i(t)$ that has the largest correlation $C(r,v_i)$ also has the largest normalized correlation $C_N(r,v_i)$. However, when the $v_i(t)$'s have different energies, the normalized correlation accounts properly for such and permits the decision to be properly based.

It is a well known fact[12], called the *Schwarz Inequality* or the *Cauchy Inequality*, that

$$|C_N(x,y)| \le 1,$$

with equality if and only if one signal is an amplitude scaling of the other; i.e. $y(t) = a\,x(t)$ for some constant a. This means that normalized correlation provides an absolute, rather than relative, measure of signal similarity. If $C_N(x,y) = 1$, it means that $x(t)$ and $y(t)$ differ only by an amplitude scaling, and the scale factor is positive. That is, they have the same shape. If $C_N(x,y) = -1$, then $x(t)$ and $y(t)$ again differ only by an amplitude scaling, but this time the scale factor is negative, meaning that one signal is the

[12]The proof is beyond our scope. It can often be found in linear algebra textbooks.

"flip" of the other. On the other hand, when $C_N(x,y) = 0$, $x(t)$ and $y(t)$ are about as unrelated as possible. Indeed, we say $x(t)$ and $y(t)$ are *uncorrelated* when $C_N(x,y) = 0$, or equivalently, when $C(x,y) = 0$.

Having introduced correlation, we can now completely describe a typical correlating detector. Suppose we must decide which of the signals $v_1(t), v_2(t), ..., v_M(t)$ is contained in $r(t)$. The decision statistic calculator computes and outputs $C_N(r,v_1), C_N(r,v_2), ..., C_N(r,v_M)$. The decision maker makes finds the largest of these, and outputs the corresponding decision.

Comparison of Energy and Correlating Detectors: There are some situations where energy detectors cannot be used and some where correlating detectors cannot be used. For example, energy detectors cannot be used for the "which signal" problem when the signals have the same energy, which is often the case in digital communications. On the other hand, correlating detectors cannot be used when the precise shape of the signals is not known. For example, in Marconi's original transatlantic radio transmission, the transmitted signal was generated by a spark, with no known signal shape. Clearly, a correlating detector was out of the question!

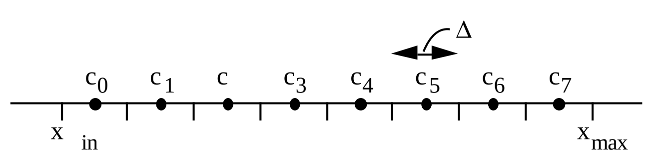
In situations where both energy and correlating detectors can be used, it is usually found that the latter performs significantly better than the former, i.e. it makes fewer errors.

C. Other Signal Processing Tasks.

There are many other signal processing tasks. Here we mention just one.

Signal Digitization: In today's world where signal processing is increasingly done by general or special purpose computers, it is important to convert signals into digital form. This involves two steps: (1) sampling, and (2) representing each sample as a binary number. Both of these steps generally involve losses, i.e. changes to the signal. Sampling is the topic of Chapter 4 and will be extensively discussed there. Converting to bits will be the subject of one of our lab assignments. However, let us describe here the most elementary method of converting samples to bits, called *uniform scalar quantization*.

With uniform scalar quantization, if we wish to represent a sample value $x[n]$ with b bits, then as illustrated below for the case that $b = 3$, we divide the range of sample values, (x_{min},x_{max}) into 2^b nonoverlapping bins of width $\Delta = (x_{max}-x_{min})/2^b$. These bins are indexed from left to right be the integers $0, 1, 2, ..., 2^b-1$, and each of these integers is represented as a b-bit binary number. For example, if $b = 3$, then $5 \Leftrightarrow 101$. Let $x_i = x_{min} + \Delta/2 + i\Delta$ denote the center of the ith bin. Now, if the sample $x[n]$ to be quantized lies in the ith bin, then we represent it by the binary representation of i, and we consider $x[n]$ to have been *quantized* to the value c_i. Note that when using this binary number in a processing task, we consider it to represent the value c_i, and must act accordingly. Actually, if the processing is done in a general purpose computer, we might convert i to binary using one of the standard conventions that are convenient for doing arithmetic, such as "two's complement".



bin index:	0	1	2	3	4	5	6	7
binary representation:	000	001	10	011	100	101	10	111

A system that does both sampling and uniform scalar quantization is called an *analog-to-digital converter.*

There are more sophisticated methods for converting samples to bits that produce many fewer bits. These are generally called *data compression* methods. Examples include JPEG image compression, MP3 audio compression, and CELP speech compression, which is used in digital cellular telephones, digital answering machines, and the like. A simple example of a JPEG like image compression is included in one of the lab assignments. Generally speaking, data compression is done in order to reduce the amount of memory needed to store a signal or the amount of time needed to transmit a signal. When the signal actually needs to be processed or played, the compressed representation must ordinarily be changed back into a representation like the one produced by a uniform scalar quantizer. This is called *decompression*.

Concluding Remarks

Having discussed these basic signal processing tasks, it should be mentioned that from now we will not focus on them in future lectures. Instead we will focus on developing tools and techniques that enable systems to perform these tasks well. In particular, much of the remainder of the course (corresponding to Chapters 5-8 of our text) will focus on methods for designing and analyzing linear filters, with the applications like signal recovery in mind. But only from time to time will signal recovery be mentioned. On the other hand, the labs will be concerned with these two principal signal processing tasks and some other tasks as well, such as signal digitization.