Lab 7 Bit Detection with the Correlation Receiver

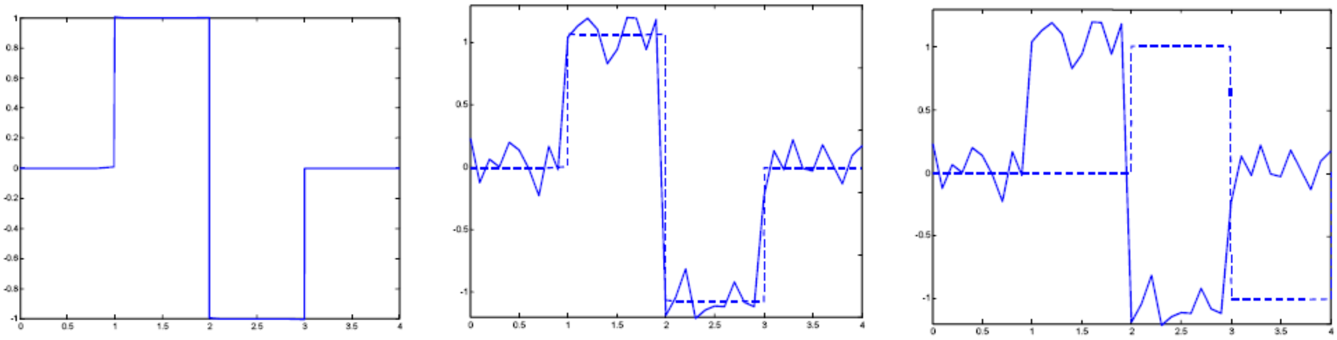
Objectives:

- Know the basic operation of the optimal matched filter and correlation receivers for detecting known/deterministic waveforms in noise.
- Implement the correlation receiver for detecting signals in noise.
- Implement the match filter for decoding bit streams received from noisy, bandlimited channels, and computing bit error rates.
- Design an experiment to determine relationship between bit error rate, bandwidth, and SNR when using the matched filter/correlation receiver.

1. Background:

Modern digital communication systems use signals to transfer information between locations from sources, such as speech, video, and text. Noise corrupts signals during transmission. This noise arises from various sources, including interference from other transmitters and thermal motion of the electrons in the transmission media. The purpose of the receiver is to accurately recover the transmitted information in the presence of noise and distortion.

The problem of recovering signals corrupted by noise has several solutions. The correlation receiver can be shown to be the best solution (optimal) for recovering a known waveform in the presence of additive white Gaussian noise (AWGN). A correlation receiver compares the received signal to a set of waveforms it expects to receive. This scenario applies directly to a digital communication problem since each bit (or bit sequence) can be encoded to a set of distinct waveforms known to both the transmitter and receiver. For example, consider the signal of Fig. 1a. If the transmitted waveform is corrupted by noise, the received signal may look like the solid-lined plots of Figs. 1b and c. A correlation operation aligns the received signals values with a template (shown in the dashed lines of Figs. 1b and c) and takes the sum of products between the received signal and template over the duration of the waveform template. The correlation is greatest when the template is aligned with the received signal, as shown in Fig. 1b. A lesser response occurs as the misalignment increases. Fig. 1c shows a misalignment that results in the minimum response (largest negative number). The correlation receiver often uses signals that are zero-mean to exploit the cancellation between positive and negative values in the correlation sum.



(a) (b) (c)

Figure 1. (a) Ideal transmitted waveform. (b) Matched alignment with noisy received waveform for maximum correlation receiver response. (c) Mismatched alignment with noisy received waveform for minimum correlation receiver response.

A **correlation receiver** therefore parses the signal into intervals that are synchronized with the bit intervals, and integrates the product of the received signal and template over the interval to produce correlation values. The correlation values become detection statistics used to decide the most likely bit value that was transmitted for each interval. In the case of binary signaling (0 or 1) there are only 2 possible symbols, so 2 correlation receivers would operate in parallel corresponding to waveforms associated with each bit. The detected bit corresponds to the channel with the greatest correlation value. The synchronization of the segment intervals requires that both the source and receiver have the same clock signal (or an alternative way to synchronize) before applying the correlation receiver, since the alignment over the correlation interval affects the result. An example of this receiver is shown in Fig. 2, where $f_0(t)$ is the template for the 0 bit, $f_1(t)$ is the template for the 1 bit, and *T* is the synchronized bit interval. Every *T* seconds the input is correlated with the template and sampled to obtain the detection statistics. In this case the decision rule is simply to decide on the bit whose template has the best match (largest value) to the incoming signal. A bit error occurs when noise or bandwidth limitations result in an incorrect decision. For a given test or experiment, **the numbers of errors per bit is referred to as the bit error rate (BER), and the expected value of the BER is the probability of error.**

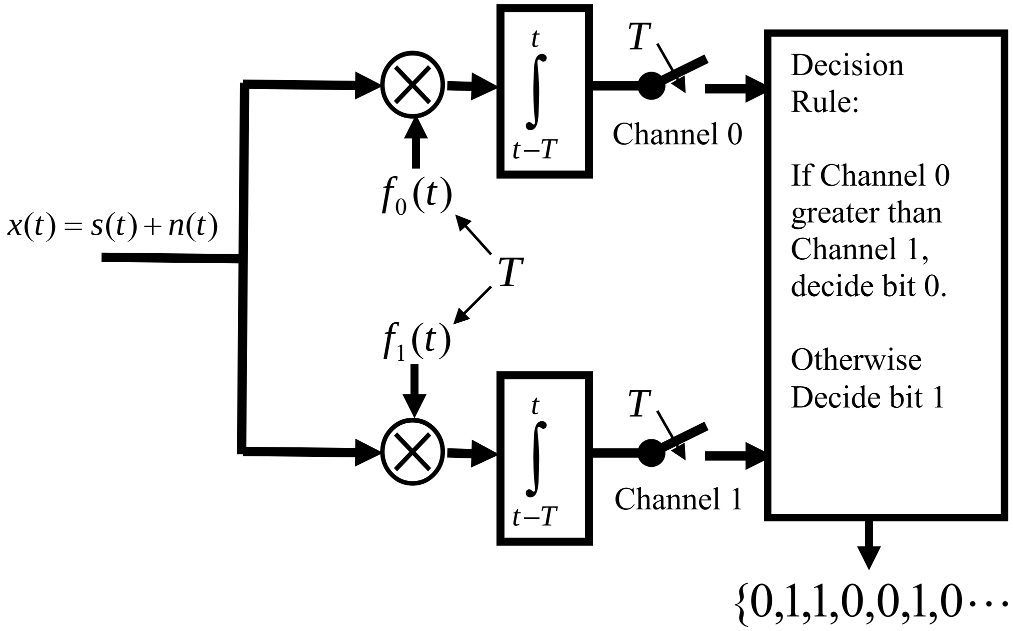


Figure 2. Correlation receiver synchronized to waveform intervals for bit sequence detection.

When a physical event emits signals apart from a synchronized clock (such as a blood pressure drop, or an echo from active sonar), the receiver must detect the presence of signal as well as estimate when the signal was received. This is typically the case for pulse-echo systems, where the time of the received echo depends on the distance between the target and receiver (which can vary as the target moves through the space of interest). The detection of the echo signal indicates the presence of a target, and the time at which the echo returns indicates the distance between the target and receiver. In this case, intervals for correlation cannot be predetermined. Therefore, the signal template is *slid* continuously (or in very small increments relative to the signal interval) over the received signal while applying the correlation operation with the template. This produces a continuous output from which decisions are made on whether a target is present. This is typically done with a simple threshold. If there is a match with the signal of interest at a particular alignment, the output will exceed typical values corresponding to the no-signal case. An example of the correlation filter implemented as a matched filter for this case is shown in Fig. 3. The signal of interest is a tapered sine wave. The top set of 3 waveforms in Fig. 3 show the signal, added noise, and the continuous sequence of detection statistics for the template that matches the signal. The lower 3 waveforms show the case when no signal is present (noise only). Note that detection statistics near 0.2 seconds reach a maximum for the case when a signal is present. This value is at least one order of magnitude greater than any of the outputs for the noise only signal. If accurate statistics of the noise fluctuations are available, the threshold can be set to achieve a specific false-alarm rate. In general, there is a trade-off between lowering the false-alarm rate (the likelihood that noise is detected as the signal – false detection) and raising the detection rate (likelihood that a target will be detected when present – true detection). A high threshold results in lower false-alarm rates but also lower detection rate, and vise versa.

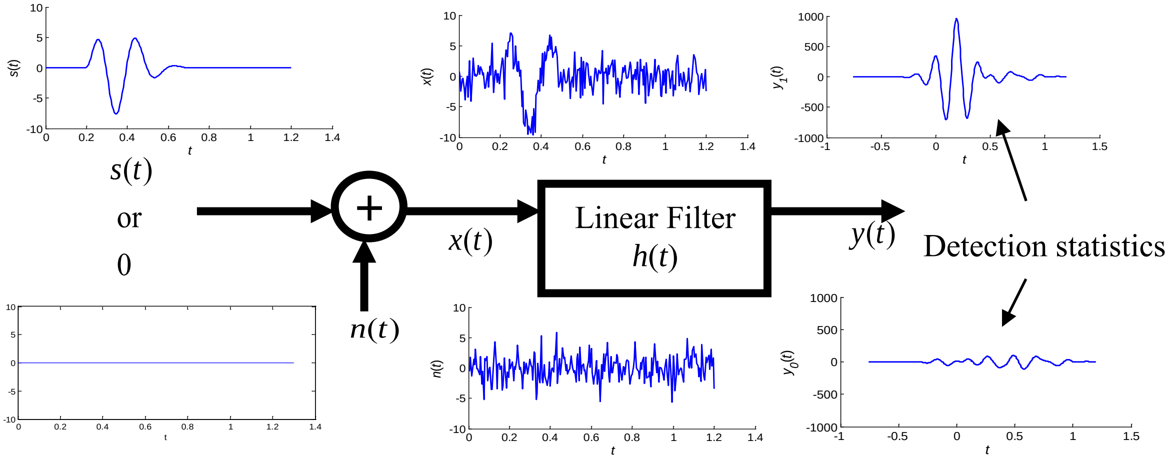


Figure 3. Correlation receiver for asynchronous signal detection. The filter impulse response, *h*(*t*), corresponds to a reverse image of the signal template, *s*(*t*), resulting in a matched filter implementation of the correlation receiver. Correlation output examples are provided for the case of noise only (*n*(*t*)) and signal plus noise.

For the following lab exercises, the noise signal, $n(t)$, will be modeled as a zero-mean Gaussian process that is uncorrelated with the signal (i.e. the AWGN channel model). This model is reasonable for many types of physical processes. The receiver works on the principle that the integration over time, averages out the noise, since the noise is not correlated with the signal template.

The correlation receivers, as shown in Fig. 2, are difficult to implement with a circuit, since analog multipliers are complicated. An easier implementation of this receiver, which gives the same result, is the matched filter. This filter works on the principle that the linear filter can be designed to perform the same operation as the correlation's multiply-and-integrate function.

Implementing Correlation Receivers as Matched Filters

For the correlation receiver, a template of the signal to be detected is *slid* across the incoming signal, where it is continuously multiplied and integrated with the section of the incoming signal that it overlaps. This can be expressed in terms of the correlation integral:

$$c(t) = \int_{-\infty}^{\infty} x(\tau) f(\tau - t) d\tau = \int_{-\infty}^{\infty} x(\tau - t) f(\tau) d\tau \qquad (1)$$

where *x*(*t*) is the incoming data, and *f*(*t*) is the template for the signal of interest. Note that it does not matter mathematically if the template is delayed relative to the data or vice versa (this commutative relationship is property of the linear convolution integral as well).

The correlation operation described in Eq. (1) slides the template over the data, estimating the degree of the match between the incoming signal and the template. **This operation is similar to convolution with a subtle difference;** convolution reverses the system impulse response (in time) prior to performing the multiplication and integration over the sliding window. Therefore, if a template is flipped (or time-reversed) and treated as a filter's impulse response, then the filter operation reverses it back to the original template and effectively performs a correlation. The resulting operation, referred to as matched filtering, has an identical outcome to the correlation receiver.

To further illustrate the relation between the correlation receiver and matched filter, consider the linear filter expressed as a convolution integral:

$$c(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau = \int_{-\infty}^{\infty} x(t-\tau)h(\tau)d\tau \tag{2}$$

where $h(t)$ is the impulse response of the matched filter, and $x(t)$ is the input. In order to use this as a matched filter for detecting bits from their line codes, $h(t)$ can be selected as a time-reversed version of the expected waveform shape for a logical bit and shifted by the bit period T to ensure that the system is causal:

$$h(t) = s_b(T-t) \text{ for } t \text{ in } [0,T] \tag{3}$$

where $s_b(t)$ is the template representing the line code for the bit to be detected. The convolution integral or matched filter on waveform $x(t)$ becomes:

$$c(t) = \int_{o}^{T} x(\tau)s_b(T-t+\tau)d\tau \tag{4}$$

Now, if we sample $c(t)$ at time $t=T$, the convolution integral becomes

$$c(t)\big|_{t=T} = \int_{o}^{T} x(\tau)s_b(\tau)d\tau \tag{5}$$

which is the same mathematical operation performed by the correlation receiver! Thus, the filtering operation of Eq. (2) with the sampling operation matches the correlation receiver output.

2. Pre-Lab Questions

1. Compute the correlation integral between a sine and cosine wave with unit amplitudes and frequency 1 kHz. Integrate over 1 full period (show your work).
2. Compute the correlation integral between 2 identical sine waves with unit amplitudes and frequency 1 kHz. Integrate over 1 full period (show your work).

3. (a) Compute the correlation integral between a sine wave with unit amplitude and frequency 1 kHz with a unit-amplitude sine at frequency 1.2 kHz. Integrate over 1 full period for the 1kHz wave. Determine an interval where an integer number of periods for both waveforms are included and integrate over this interval.
   (b) Compute the correlation integral between a sine wave with unit amplitude and frequency 1 kHz with a unit-amplitude sine at frequency 2 kHz.
   Integrate over 1 full period for the 1kHz wave.
4. Write a Matlab function to compute bit error rates. The function inputs should be 2 vectors of the same size. The first one is a vector of the original or transmitted bit sequence (a vector of 1's and 0's) and the second one is the received bit sequence with possible differences from those transmitted. These differences represent bit errors. The output should be a scalar indicating the error rate. Hand in a hard copy of the commented code.

3. Laboratory Exercises:

For each problem below, code needs to be developed to implement the experiments. Make sure these are commented and included in an appendix for the lab report.

1. Simulate a digital receiver with matched filter using a sampling rate of fs=64kHz, a bit rate of br =4kb/s, and a channel bandwidth of bw=20kHz. Let the decision rule be to decide on the bit corresponding to the template with the highest matched filter response. Create a plot of BERs vs. SNR to observe the bit error rate as a function of SNR. The plot should range from the SNR at which the BER is effectively zero (or close to it $<10^{-4}$) and to the SNR where the bit error rate exceeds 20% (> .2). You should select enough intermediate SNR values so the plot looks reasonably smooth. Plot the error rate axis using a log base-10 scale. Compute the bit rates using at least 10,000 bits per SNR level. Perform this experiment for the following line codes:
   1) bipolar nrz
   2) bipolar rz
   3) Manchester
   4) bipolar Nyquist

   Present the plots of SNR vs. BER comparing the performance of different line codes to the TA. Indicate the codes that show robust performance in the presence of noise.

2. Repeat the experiment in the previous step, however this time fix the SNR at 21 dB and reduce the channel bandwidth starting with a maximum of 20kHz. Plot the bit error rate as a function of bandwidth. Reduce the bandwidth until the bit error rate exceeds 20%. **Show the TA the resulting plots and indicate** the line codes that show the most robust performance to bandwidth limits.

3. For this exercise a data set was created consisting of 3-second signals that were generated by corrupting a square wave with increasing levels of AWGN. The signals were sampled at 10kHz. The square wave was generated asynchronously from the Manchester line code for bit symbol 1, with period 0.01 seconds, and occurred at various places within the 3 second received signals. The data for this
(sqwaveINnoise.mat)
signal was stored in a mat file found at: http://kstio.com/index.php?p=1_4_digital-signal-processing

Download file and load it into Matlab's workspace. Once loaded, type the "whos" command to display the following information:

```
Name        Size        Bytes       Class    Attributes
  fs        1x1            8        double
  sigmat    12x30000       2880000  double
  snra      1x12           96       double
  tax       1x30000       240000    double
```

The scalar variable *fs* is the sampling rate of the received signals stored in rows of matrix *sigmat*, in which each row contains a square wave signal embedded in the noise. Each row has noise of increasing power levels. The SNR values corresponding to each row is stored in the vector *snra*, and the vector *tax* is the time axis in seconds associated with each row of the signal matrix *sigmat*.

In this exercise you must determine where the square wave is located for each row of *sigmat*. For some of the higher SNR values this can be done by simply plotting the received signal and inspecting it visually; however you should apply a matched filter and observe the filtered output as well as the original. You can use the zoom feature on the plots and data curser to **find the value in seconds where the beginning of the square wave is located**. Show the TA your filtered plots and indicate the locations detected for each level. For the lower SNRs it will not be possible to confidently indicate where the target signal is located. **Indicate to the TA the highest SNR level associated with undetectability in your results.**

4. Design an experiment to the show the relationship between BER and SNR and bandwidth for any one of the line codes when using the matched filter / correlation receiver. This experiment should result in a family of plots over the critical range (i.e. where distinguishable differences are happening) where the y-axis is BER and the x-axis is bandwidth. Each member in the family of plots will be the result of a different SNR value. Present the plots in a way that the precision of the experiment is understood (i.e. that differences in the plots are not just the result of experimental variability). Create a report for this experiment as done for labs 1 and 2.