EE416 Final Project 2016

Professor James A. Ritcey
Department of Electrical Engineering
University of Washington
Seattle, WA 98195
206-543-4702 (voice) 206-543-3842 (fax)
ritcey@ee.washington.edu

November 30, 2016

1 EE 416 Project Rules

1. The Final Project for EE 416 is a group 2-person project. Each group should have two people in it. Both are to share equally in the work, and each will receive the same project grade, except for extraordinary circumstances. If you encounter group problems, talk to me or the TA.
2. I expect that each group does its own work, without collaboration with anyone. Cite any sources you use, including web pages. No plagiarism in any form is acceptable including matlab code, copying, help from others beyond the TA and myself.
3. The report should be concise and complete. Due date and submission information is on our website.

2 Introduction

The matched filter is the main signal processing structure for optimal reception of signals in additive white Gaussian noise. Since its origins, it has been the main impetus for spread spectrum signaling in radar, sonar, and digital communications. This was first used for military applications, but was later found to be important in commercial applications. These involve uncoordinated transmission over unlicensed frequency bands. This is the band of interest for Wi-Fi 802.11 systems. Spread spectrum has also been important for coordinated multiple access in cellular. Code division multiple access or CDMA uses a family of spread spectrum signals to modulate information on the base to mobile down-link in cellular. The US company, Qualcomm holds many of the patent rights to this technology and CDMA has proved to be the dominant cellular technology for several generations. Most military systems use CDMA. Radar systems also use spread spectrum codes, as well as preambles in comm for signal acquisition. These applications are closer to our project than CDMA, but the theory is the same.

In this project, we investigate the performance of spread spectrum using discrete time models of the transmission, channel, and received signal. You are required to follow the outline given in this problem statement, develop supporting MATLAB code, and present an individual report on your investigations.

3 Background: Matched Correlator

A matched correlator is used to process the received signal

$$\mathbf{r} = A\mathbf{s} + \mathbf{w} \tag{1}$$

where $\mathbf{r}$ is the signal, $A > 0$ and is unknown to the rcvr. The signal energy is $E = \mathbf{s}'\mathbf{s}$. We take all vectors to be column vectors. The theory is described in Appendix A. The additive noise is AWGN, additive white Gaussian noise, $\mathbf{w}$ in which each sample is iid with common pdf $N(0, \sigma^2)$. During the derivation we will normalize this to $N(0, 1)$ and work in terms of the SNR $S := A^2E/\sigma^2$. This is written in dB as $SdB = 10\log_{10} S$,because this is an energy ratio.

The correlator is a linear operation, essentially a dot or inner product. It uses a weight function $\mathbf{h}$ and takes input $\mathbf{r}$. This are "dotted "to give

$$V = \mathbf{h}'\mathbf{r} \tag{2}$$

When the correlator is matched to the received signal, we set

$$\mathbf{h} = k\mathbf{s}. \tag{3}$$

You can choose to use the unit vector,

$$\hat{\mathbf{h}} = \mathbf{h}/|\mathbf{h}| \tag{4}$$

Here

$$|\mathbf{h}|^2 = \mathbf{h}'\mathbf{h} = E_h = \sum_n |h_n|^2 \tag{5}$$

For a unit vector, this energy will be one. What is the energy in $\mathbf{h}$? It is simply the Euclidean length squared. This is because $\mathbf{h}$ is deterministic. When we have a random signal, we must follow the length-squared with expectation.

3.1 Task 1

Derive the distribution of V and $v = V/\sigma_V$, where σ_V is the standard deviation of RV V. Use the parametrization in terms of S. Simulate this process in matlab using $Nt = 1e+03$ trials. Show by overlaying the histogram and the theoretical pdf that you have an excellent match. Use a chi-square goodness-of-fit test to quantify the match. Since you control the simulation experiment, the fit should be excellent. You may have to read up on the Chi-square Test. However, it can be implemented in Matlab quite easily and was in an earlier assignment. Describe fully what you are simulating. Do not expect me to understand your approach from reading of the Matlab code. *Use Diagrams.*

The output of the correlator is compared to a threshold v_0. The detection probabilities are defined by 2 threshold crossing events

$$\begin{aligned} P_{fa} &= P\{V > v_0|H_0\} && (6) \\ P_d &= P\{V > v_0|H_1\} && (7) \end{aligned}$$

These can be worked out in terms of the Gaussian tail probability

$$Q(x) = \int_x^\infty \exp(-x^2/2)\frac{dx}{\sqrt{2\pi}} \tag{8}$$

In MATLAB you might use

$$Q(x) = \frac{1}{2}\text{erfc}(x/\sqrt{2}). \tag{9}$$

3.2 Task 2

For **your system** *scaling*, Construct a table of threshold values giving the pairs (P_{fa}, v_0). Set

$$P_{fa} = 10^{-x}, \quad x = 1, 2, 3, 4.$$

Construct some type of a graphical display of this table and comment on the increase in v_0 with the decrease in P_{fa}. Using these thresholds, **derive** P_d **and plot** P_d versus S in dB over a reasonable range, for $x = 2, 4, 6$ from above. Of course, use the appropriate semilogy format. Explain the offsets of the plots, that is, why each plot is roughly a shift or offset from the others. These results only apply to a fixed scaling, changing the variance will require changing the threshold.

4 Matched Filtering

The correlator is the simplest implementation since it requires only a single multiplier and reads out h_n from RAM. The outputs of the multiplier $h_n r_n$ can be accumulated, once they are determined. A filter is a more complicated (ie, uses more operations) structure because it requires parallel multiplication. Recall that an FIR filter uses impulse response h_n (NOT the same as the correlator **h**, but closely related) and responds to input x_n to put out y_n where, for a causal h_n, x_n,

$$y_n = h_n \star x_n = \sum_{k=0}^{n} h_{n-k} x_k \tag{10}$$

This can be implemented using `filter.m`

An alternative to the correlator structure is to process the input r_n and to put out

$$y_n = h_n \star r_n, \tag{11}$$

and to compare this to the threshold. The advantage of the filtering structure appears when we consider a signal of unknown Time of Arrival (TOA). In this case, under H_1 we model

$$r_n = A s_{n-n_0} + w_n \tag{12}$$

The same assumptions hold as with the correlator; the only modification is the unknown time of arrival ℓ. The convolution structure lets us examine all possible TOAs, and to detect the presence of multiple signals. However, the signal shape, along with its energy, becomes important.

A matched filter h_n is designed for signal s_n according to

$$h_n = s_{N-n}, \tag{13}$$

Where N is the length of s_n. In effect this time reversal is introduced to compensate for the flip introduced by the convolution. The output is called a correlation, as opposed to a convolution.

$$s_n \star s_{-n} = \sum_{k} s_k s_{k+n} \tag{14}$$

Notice that this output is of length $2N - 1$, since the convolution $h_n \star r_n$ is of length $L_h + L_r - 1$, where L_x denotes the length or time duration of x.

The signal **s** is called the code, or signature. From a transmission efficiency point of view, it is highly preferred that this signal be constant envelope. That is it should be a bi-phase sequence. This sequence of $\{+1, -1\}$ ideally has a large peak and low peak sidelobes. The best codes are the Barker codes, and these are included in my test program for lengths $N = 11, 13$. No larger Barker code is known. You may ask why a constant amplitude signal is used, ie $\{+1, -1\}$. The reason is that the transmitter amp can be driven in into saturation with maximum gain.

The random code sequence looks to be a nice spread spectrum signal. Its wide bandwidth is compressed by the matched filter into a peak with energy E. The time sidelobes are low. Of course, both the transmitter and receiver must known s_n, in order to implement the matched filter at the receiver. More problematic is the fact that the amplitude is not constant, and the large peaks in s_n can cause dynamic range problems. So, it is preferable (really required) to use a antipodal sequence. This is a sequence of binary $\{+1, -1\}$. These sequences need to be found from a computer search or otherwise.

4.1 Task 3

Please construct a plot of the (auto)-correlation function of the following 3 sequences. The encoding I use is that 0 represents +1 and 1 represents −1.

1. The alternating series of length $N = 11$. This is

$$[01010101010]$$

2. The $N = 11$ chip Barker code (used in WiFi 802.11). This is

$$[01001000111]$$

3. The $N = 13$ chip Barker Code. This is

$$[1010110011111]$$

Determine the PSL, the peak sidelobe level. Why are the Barker codes so good? The correlation function can be computed using `xcorr.m`, which implements

$$R_{sr}(k) = \sum_{\ell} s_\ell r_{\ell+k}$$

Typically, the sum ranges over all nonzero values of r, s. This is the cross correlation between s and r. You need to use the autocorrelation R_{ss}. Just set $r = s$.

4.2 Task 3.5

Barker codes go only up to length 13, beyond that there are no perfect codes. However, you are asked to find a longer length code that has low sidelobes. Search wikipedia "Barker code" to find a listing of all known Barker codes.

We can create longer good, not perfect, codes by "'placing one code into another". For example, a code $[c0c1]$ is placed into $[s0s1s2]$ to create [$s0 * c0$ $s0 * c1$ $s1 * c0$ $s1 * c1$ $s2 * c0$ $s2 * c1$]

Using Wikipedia's listing of Barker codes, **create longer codes** by placing the lengths [2 3 4 5 7 11 13] into a Barker-13 code. Create a table that provides the code Identification, mainlobe maximum in dB, the average sidelobe, and the peak max sidelobe; all in dB re the max value. To compute these statistics: consider the sidelobe levels in absolute value, treat them as iid, then simply calculate mean, variance, etc Discuss the tradeoffs and which code you might recommend for a particular application. Vehicular radar is a good current one. These codes aren't perfect, in particular, how they handle distortion due to Doppler or multipath can be an issue.

4.3 Task 4

For this you need to construct a matched filter decoder that reads an input files (You download this) and detects 10 signals. Each signal is a Barker 13, delayed to a random starting time. Each group will receive a different dataset; these are marked by Group Number. **You will need to determine the starting time of the 10 signals.**

Provide a block diagram and description of your system, along with the MATLAB code.

To detect a signal you must determine its starting time. You know the code, and you know the signal, antipodal Barker-13, the observation is noisy (Gauss noise) and you need to apply the MF. The MF output simply "'peaks or compresses" the received signal. You still need to threshold to detect and find the start time. You will miss signals and detect extras (false alarms). However, you are graded on how accurate you are. Big Hint: Test on your own signals First! That way you know the performance at high SNR. I am looking for the starting times of each Barker-13.

I will provide an additional common dataset that you can try your algorithm on.

4.4 Task 5

The matched filter concept is used in many other applications. Find a application not discussed in the text and provide a one-page summary including references. You may find this in radar, communications, optics, physics, biomedical signal processing, etc.

5 Appendix A: Statistical Theory of the Matched Correlator

A common problem in statistical digital processing is detection. Applications include the demodulation of a digital signal in communications, or deciding for the presence of a signal in a radar or sonar. Some key attributes of the problem include:

1. The uncertainty is very limited – either the signal is present or absent. The signal shape, but not its amplitude, is assumed to be known.

2. The noise and interference can be described statistically in terms of its ACF or PSD.

The problem can be posed and solved in several ways - in discrete or continuous time, with finite duration observations, or with vector observations. The latter is the easiest to formulate and solve; the others can always be reduced to this situation.

5.1 Hypothesis Testing Problems

The framework is that of a Hypothesis Test. There are two states of nature, denoted H_0 or H_1. Typically, the null hypothesis is reserved for the status quo, while H_1 denotes the alternative, usually a rare event. In digital communications, the situation is usually more symmetric. The degree of likelihood may be represented by prior probabilities, $P_0 + P_1 = 1$, where P_k is the probability that H_k is true, *before observing any data.* However, in this work we will assume these are unknown, and apply the Neyman-Pearson criterion, maximize the Detection Probability subject to a False-Alarm constraint.

We observe a vector $\mathbf{x}$. The following model will be used, where all vectors are $N \times 1$ columns.

$$H_0 \quad : \quad \mathbf{x} = \mathbf{w} \tag{15}$$

$$H_1 \quad : \quad \mathbf{x} = A\mathbf{s} + \mathbf{w} \tag{16}$$

Here, $A > 0$ is an unknown amplitude, $\mathbf{s}$ is a completely known signal, and $\mathbf{w}$ is a vector of AWGN. What is uncertain is which hypothesis is true. The problem can be generalized to demodulate information conveyed by unknown symbols A.

To solve this problem, we need to process $\mathbf{x}$ and make a decision. To do this optimally requires a criterion, we'll use maximum probability of detection, P_d, subject to a prescribed P_{fa}.

The solution consists of signal processing; first we filter $\mathbf{x}$, then we compare the filtered result to a threshold. When the threshold is exceeded an alarm is declared and we decide for H_1. We will skip the derivation of this structure (use the likelihood principle), and focus on designing the filter using a maximum SNR criterion. Here the filter is FIR, and can be viewed as a dot product.

The processing is a two-step procedure: First, sample the output of a linear filter

$$V = \mathbf{c}'\mathbf{x} \tag{17}$$

Second, compare this to a threshold, V_0, and decide for H_1 when

$$V > V_0. \tag{18}$$

Otherwise, decide for H_0. Both the filter and the threshold must be determined to specify the signal processing. This is the so-called correlator structure, the formation of V above.

5.2 Max SNR Filter Design

Consider the following design problem. Find a vector $\mathbf{c}$ to maximize the SNR in

$$V = \mathbf{c}'\mathbf{x}. \tag{19}$$

Since all vectors are compatible columns, V is a scalar. Since the processing is linear, SNR can be defined as the ratio of desired signal power to undesired noise power,

$$\text{SNR} = \frac{A^2 E\{(\mathbf{c}'\mathbf{s})^2\}}{E\{(\mathbf{c}'\mathbf{w})^2\}} \tag{20}$$

The numerator is deterministic, and contains the filter $\mathbf{c}$ and the known signal shape $\mathbf{s}$. The denominator must be evaluated, and we find

$$E\{(\mathbf{c}'\mathbf{w})^2\} = \mathbf{c}'E\{\mathbf{w}\mathbf{w}'\}\mathbf{c} \tag{21}$$

This involves the $N \times N$ autocorrelation matrix

$$\mathbf{R}_W = E\{\mathbf{w}'\mathbf{w}\} = P_w\mathbf{I}. \tag{22}$$

This is because for WGN, the correlation matrix is diagonal, with the average power, P_w down the diagonal.

5.2.1 Applying The Cauchy-Schwartz Inequality

We seek a solution to the following optimization criterion.

$$\max_{\mathbf{c}} \text{SNR} = \max_{\mathbf{c}} \frac{A^2}{P_w}\frac{(\mathbf{c}'\mathbf{s})^2}{\mathbf{c}'\mathbf{c}} \tag{23}$$

This optimization is carried out by using an upper bound. We find an upper bound for the numerator, known as the Cauchy Schwartz inequality, and the conditions for achieving this bound. If this can be realized, we have maximized the ratio.

The upper bound is well-known to you: a dot product is no larger in magnitude than the product of the lengths,

$$|\mathbf{c}^T\mathbf{s}| \leq |\mathbf{c}||\mathbf{s}| \tag{24}$$

or

$$|\mathbf{c}^T\mathbf{s}|^2 \leq |\mathbf{c}|^2|\mathbf{s}|^2 \tag{25}$$

which is written in terms of energies. The dot product is a maximum when the vectors are aligned,

$$\mathbf{c} = k\mathbf{x}, \text{ for some scalar } k. \tag{26}$$

Substituting, we find that the SNR is no greater than

$$\begin{aligned} \text{SNR} &\leq \frac{A^2}{P_w}\mathbf{s}^T\mathbf{s} &(27)\\ &= \frac{A^2E_s}{P_w} \text{ when } \mathbf{c} = k\mathbf{s} &(28)\end{aligned}$$

where $E_s = \mathbf{s}^T\mathbf{s}$ is the signal energy. This shows a *processing gain* (PG) equal to the signal energy as compared with the input signal.