Chapter 6

Matched Filter

In this chapter, a linear filter is found to provide a solution for three related tasks: maximizing receiver signal-to-noise ratio, estimating time-of-arrival, and estimating a channel impulse response. The linear filter in each case is a correlator known as the *matched filter*.

In the exercises, you will use use matched filtering for echo location, phase equalization, and channel estimation. The experiments are implemented at acoustic frequencies using MATLAB, rather than at radio frequencies using the PCI-5640R IF transceiver, so that sub-meter range resolution may be obtained with only a few kiloHertz of bandwidth. Note that the speed of propagation of acoustic waves in air is approximately one million times slower than that for electromagnetic waves.

6.1 Background

6.1.1 Maximizing SNR

Consider linear processing of a known signal, $s(t)$, observed in additive noise

$$r(t) = As(t) + w(t) \tag{6.1}$$

where the noise is assumed to have power spectral density $\mathcal{P}_W(f)$. The processing goal is to maximize the output signal-to-noise ratio at time t_0. The filtering must balance suppression of the noise and emphasis of the signal. By linearity, the filtered output

$$y(t) = y_s(t) + y_w(t)$$

has components due to the signal and to the noise. The *matched filter* maximizes the SNR at time $t = t_0$ and is given by

$$H(f) = \frac{S^*(f)}{\mathcal{P}_W(f)} e^{-j2\pi f t_0} \tag{6.2}$$

If the noise has a flat power spectral density (white noise), then the matched filter has impulse response

$$h(t) = s^*(t_0 - t). \tag{6.3}$$

Equation 6.3 reveals that, for white noise, the filter's impulse response is a scaled time-reversal of the known waveform, and is hence "matched" to the waveform (North, 1943).

Proof The filter output at time t_0 is the inverse Fourier transform of $H(f)S(f)$ evaluated at time $t = t_0$,

$$y_s(t_0) = \int_{-\infty}^{\infty} H(f) S(f) e^{j2\pi f t_0} df.$$

The average power of the noise output is

$$\mathrm{E}\{|y_w(t_0)|^2\} = R_{y_w}(0) = \int_{-\infty}^{\infty} \mid H(f) \mid^2 \mathcal{P}_W(f) df.$$

The signal-to-noise ratio to be maximized is therefore

$$\mathsf{SNR} = \frac{\int_{-\infty}^{\infty} H(f) S(f) e^{j2\pi f t_0} df}{\int_{-\infty}^{\infty} \mid H(f) \mid^2 \mathcal{P}_W(f) df}.$$

The maximizer is found with the aid of the Cauchy-Schwarz inequality,

$$\left| \int_{-\infty}^{\infty} a(x) b(x) dx \right|^2 \leq \int_{-\infty}^{\infty} \mid a(x) \mid^2 dx \int_{-\infty}^{\infty} \mid b(x) \mid^2 dx \tag{6.4}$$

where equality is achieved if and only if $a(x) = K b^*(x)$ for some constant K. To proceed, make the assignments

$$a(x) \leftrightarrow H(f)\sqrt{\mathcal{P}_W(f)}, \qquad b(x) \leftrightarrow \frac{S(f) e^{j2\pi f t_0}}{\sqrt{\mathcal{P}_W(f)}}$$

to observe

$$H(f)\sqrt{\mathcal{P}_W(f)} = \frac{K S^*(f) e^{-j2\pi f t_0}}{\sqrt{\mathcal{P}_W(f)}}.$$

from which Equation 6.2 follows. $\blacksquare$

The implementation of the matched filter is a correlator:

$$y(t_0) = \int_{-\infty}^{\infty} r(\tau)h(t_0 - \tau)d\tau = \int_{-\infty}^{\infty} r(\tau)s^*(\tau - t_0)d\tau \qquad (6.5)$$

For sampled-data processing, we have

$$y[k] = \sum_{n=-\infty}^{\infty} r[n]s^*[n-k]. \qquad (6.6)$$

6.1.2 Time-of-arrival estimation

For time of arrival estimation, we seek to compute the time at which a known waveform with unknown gain arrives at a sensor, such as a microphone, hydrophone, or antenna. The signal model is familiar:

$$r(t) = As(t - t_0) + w(t) \qquad (6.7)$$

For additive white Gaussian noise (AWGN) $w(t)$, the maximum likelihood estimate (Poor, 1994) of the arrival time, t_0, is computed using the absolute value of the output from a matched filter:

$$\begin{aligned} y(t) &= \int r(\tau)s^*(\tau - t)d\tau \\ \hat{t}_0 &= \arg\max_t \mid y(t) \mid . \end{aligned} \qquad (6.8)$$

Further, an estimate of the unknown amplitude A is also produced by the matched filter:

$$\hat{A} = \frac{y(\hat{t}_0)}{\int \mid s(\tau) \mid^2 d\tau} \qquad (6.9)$$

That is, A is revealed from the ratio of the observed peak value to the known noiseless autocorrelation peak from $s(t)$. For complex-valued signals, A contains both a gain and phase.

In radar and sonar, the processing in Equation 6.8 is known as *pulse compression*, because a signal $s(t)$ of some duration T seconds produces a sharp peak at the receiver.

6.1.3 Timing recovery

In many communication systems, digital data are transmitted in packets, and a receiver must locate the start of each packet of data. This is known as *frame synchronization.* A receiver can estimate frame timing using a marker sequence and the matched filter. The time reference is located using the peak absolute value of the cross-correlation between the known marker sequence and the received marker-plus-noise.

Thus, a good marker sequence must have a very sharp autocorrelation peak, so that the cross-correlation peak is easily identified in noise. A maximum-length pseudonoise sequence (m-sequence) of plus and minus ones is the peakiest possible autocorrelation sequence:

$$R_c(k) = \frac{1}{N}\sum_{n=0}^{N-1} c[n]c^*[n-k] = \begin{cases} 1, & k = N \\ \frac{-1}{N}, & k \neq N. \end{cases} \tag{6.10}$$

The m-sequences are easily constructed for lengths that are a Mersenne number, i.e., a number of the form $N = 2^n - 1$ for some positive integer n.

6.1.4 Channel estimation

For a marker signal $c[n]$ with good autocorrelation properties, the autocorrelation is approximately a scaled delta sequence. From Equation 6.10 we have

$$\frac{1}{N}\sum_{n} c[n]c^*[n-k] = \left\{ \begin{array}{ll} 1, & k = N \\ \frac{-1}{N}, & k \neq N. \end{array} \right\} \approx \delta[n-N]. \tag{6.11}$$

With this observation and the associative property of convolution, we can use the correlation receiver and a marker signal to estimate a channel impulse response, $h[n]$. Let $r[n] = h[n] \star c[n] + w[n]$ be a noisy received version of the marker sequence $c[n]$ distorted by convolution with the channel impulse response, $h[n]$. The correlator at the receiver yields

$$\begin{aligned} \frac{1}{N} r[n] \star c^*[-n] &= \frac{1}{N}\left\{h[n] \star c[n] + w[n]\right\} \star c^*[-n] \\ &= h[n] \star \left\{\frac{1}{N} c[n] \star c^*[-n]\right\} + \frac{1}{N} w[n] \star c^*[-n] \\ &\approx h[n] \star \delta[n-N] + \frac{1}{N} w[n] \star c^*[-n] \\ &\approx h[n-N]. \end{aligned} \tag{6.12}$$

6.2 Exercise: Echolocation

In this exercise, you will use the matched filter to implement a simple echolocation device. Use MATLAB for transmitter and receiver processing; an audio speaker will serve as transducer for transmission, while a microphone will provide the transducer on receive.

1. Write a short MATLAB program to create a DSBSC-AM transmitter. To speed you on your way, below is an outline for your code, with suggestions for useful MATLAB commands. Using a sampling rate of 44100 Hz, use `pn=sign(randn(1,Np))` to create a pseudorandom sequence of ± 1. The pseduorandom noise-like sequence is a simple way to generate a broad-band signal with good autocorrelation properties. Choose `Np` to achieve a duration of approximately 400 msec. Use a "seed" with the random number generator so that your results are repeatable. The integer given as a seed sets the state of the pseudorandom number generator so that identical values are generated at each repeated execution of the command. (For more on the use of a seed, search `randn` at the MATLAB online help page.) Lowpass filter the pseudonoise (pn) sequence to compute a message `m` having one-sided bandwidth of 4000 Hz; a filter order of 50 to 100 should suffice. Modulate the resulting baseband waveform to a carrier frequency $8000 \leq f_c \leq 17000$. Save the message `m` so that it may be used at the receiver. The commands `pause` and `wavplay` can be used to play the acoustic signal through the speaker.

```
% Echo Location Lab, Tx
% ECE508

%% define parameters

%% Make message sequence to share at Tx and Rx
%Useful commands:  firls, conv, save
randn('seed',myNum); %myNum is an integer seed
pn = sign(randn(1,Np));%Np is length of pn sequence


%% quadrature amplitude modulate
```

```
%% plot the autocorrelation of the baseband message
% and plot the baseband spectrum
figure;
plot([-length(m)+1:length(m)-1]*Ts*1000,abs(xcorr(m,m)));
title('Autocorrelation','fontsize',14);
xlabel('time (milliseconds)','fontsize',14);
figure;freqz(m,1,4096,fs);
title('spectrum of transmit signal','fontsize',14);

%% Prompt and transmit
display('Ready to transmit: hit enter');
pause;
wavplay(s/max(abs(s)),fs);
%note scaling to [-1,1] to avoid clipping
```

2. Create a similar program for your quadrature receiver. To speed you on your way, below is an outline for your code, with suggestions for useful MATLAB commands. Define parameters identical to those used at the transmitter; load the transmit waveform, `m`, from memory, and use `wavrecord` to record an acoustic waveform. Plot the received waveform. Compute and plot the cross-correlation of the received signal and the transmit signal, `m`. For the cross-correlation plot, label the horizontal axis in milliseconds, and place $t = 0$ at the location of the maximum absolute value. (Use MATLAB `help` for assistance with any MATLAB commands.)

   In Windows, use `Volume >> options >> properties` to insure that the microphone is "selected." so that sound may be recorded.

```
% Echo Location Lab, Rx
% ECE508

%% define parameters
%useful command:  load

%% Prompt, then record
display('Ready to record: hit enter.')
pause
r = wavrecord(RecordTime*fs,fs);% Sample and record
```

```
r = r(:)';%force received signal to be a row vector

%% plot the received signal

%% quadrature demodulate and implement matched filter
%useful commands:  fliplr, conv, exp, conj, max, abs

%% plot the correlation output
```

3. Place a speaker and microphone on lab stools approximately 30 cm apart and approximately 60 cm from a wall (or large cabinet). Face the microphone and speaker directly towards the wall, rather than towards each other. Measure distances using a tape measure.

4. Execute your receiver from one MATLAB command window, and execute your transmitter from a second MATLAB command window.

5. Plotting the absolute value of the crosscorrelation at the receiver, define $t = 0$ at the first correlation peak and label the arrival times of other prominent peak(s). Using 343 m/sec as the approximate speed of sound in air, compute the *round-trip* distances corresponding to the correlation peak(s). Compare your echolocation computations with physical measurements of potential sound paths.

Question 6.1: Report ranging results.

6. Repeat twice with the microphone and speaker in new positions. Record your data and compare ranging results among the three experimental configurations.

Question 6.2: Report ranging results.

7. Append your MATLAB code to your laboratory report.

6.3 Exercise: Channel estimation

In this exercise, you will use the crosscorrelation to estimate a channel model.

1. Use your transmit and receive programs to compute an estimated impulse response and frequency response for your baseband acoustic channel. (Refer to Equation 6.12.) Plot the absolute value of the impulse response, and plot the magnitude and phase response. Provide a brief

interpretation your plots. (The command `freqz` may be useful.) Is your channel model accuarately described by a single gain and phase? Why or why not?

Question 6.3: Wideband channel model.

2. Modify your transmit and receive programs for a one-sided message bandwidth of 300 Hz. Repeat part (1).

Question 6.4: Narrowband channel model.

6.4 Exercise: Phase recovery

In this exercise, you will use the crosscorrelation to estimate a gain and phase at the receiver.

1. Modify your transmit and receive programs for a one-sided message bandwidth of 300 Hz. What is the value of the autocorrelation peak of the your noiseless signal, `m`?

2. Place the microphone and speaker directly facing each other on stools no more than 50 cm apart. Execute your transmit and receive programs, recording the crosscorrelation at the receiver.

3. What is the magnitude and phase of the crosscorrelation at the index corresponding to the peak absolute value? Call this value C_{max}. Determine the relation between the index of the correlation peak and the starting index of the transmit message embedded in the long received signal. Test in MATLAB using a very small example, such as
`v = randn(1,11);v(5:7)=[2,3,4]; conv(fliplr([2,3,4]),v)`.

Question 6.6: Start index?

4. Use the index and value of the correlation peak to both locate your message in the received signal (frame timing) and to perform gain and phase equalization on your inphase and quadrature receive channels. Verify your results by plotting the inphase and quadrature channels after equalization. Describe your method and report the plot of I & Q after equalization. In addition, overlay the original real-valued message signal with the I channel and compute the normalized root-mean-squared error (NMSE) using the MATLAB `norm` command:

$$\frac{\text{norm}(v_\text{I} - m)}{\text{norm}(m)} \times 100\%.$$

Question 6.6: Method and result. NMSE.