Chapter 9
MATLAB Problems

9.1 Simulate the transmission of a single tone message in the presence of additive white gaussian noise (AWGN) to show the effects of low pass filtering on the signal's power spectral density plot and the effects of the filter's bandwidth on $(S/N)_D$.

(a) The system with a single tone sinusoidal input has the following parameters:

(i) $A_m = 1$, and $f_m = 100$,
(ii) 1st order LPF with $W = 100$ Hz,
(iii) coefficient of randn = 2.

Plot both the noisy signal and its power spectral density. You may have to expand the amplitude scale to see the noise component. Then calculate a precise value of signal-to-noise ratio (SNR) at the destination. You should run several trials and get an average destination SNR.

(b) Repeat (a) except with $W = 500$ Hz, and then $W = 1000$ Hz. Comment on how your results are affected by the LPF bandwidth.

9.2 Repeat Prob. 9.1 except use a 5th order LPF. Comment on to what degree the order of the LPF affects the calculated value of $(S/N)_D$. How significant is the order with respect to $(S/N)_D$?

9.3 Repeat Prob. 9.1 except use a matched filter. Compare the results of the matched filter against the results of Probs. 9.1 and 9.2.

9.4 Consider gaussian monocycle pulse $x(t) = \frac{t-0.5}{100} e^{(t-0.5)^2/100}$ that has been corrupted by AWGN such that $x\_noisy(t) = x(t) + n(t)$ and $\sigma_{noise} = 0.5$.

(a) Write a program to obtain estimate $\hat{x}(t) = h(t) * x\_noisy(t)$ using a first order butterworth LPF with f_0 the half power frequency of $x(t)$. Plot $x(t)$, $x\_noisy(t)$, and $\hat{x}(t)$.

(b) Repeat (a) using a matched filter.

(c) For both parts (a) and (b) calculate $(S/N)_D$. You should do several trials to obtain an average value of $(S/N)_D$. Comment on your results.

9.5 Repeat Prob. 9.4 except with gaussian pulse $x(t) = e^{(t-1)^2/100}$.

9.6 An ad hoc method of removing noise from the output of an analytical instrumentation system is the *smoothing filter* whereby the data is filtered using a weighted average algorithm. One implementation is the processed data sample is a weighted average of the original sample and eight adjacent data points, the higher weight given to those points nearest to the sample of interest. The *smoothing* operation is a form of *interpolation*.

Consider an instrumentation system where the observed data consists of the ideal data sample plus noise or simply $y(k) = x(k) + n(k)$. The weighted average operation that estimates the ideal data point could be

$$\begin{aligned}\hat{x}(k) = {} & \frac{1}{64}y(k-4) + \frac{3}{64}y(k-3) + \frac{5}{64}y(k-2) + \frac{7}{64}y(k-1) \\ & + \frac{32}{64}y(k) \\ & + \frac{7}{64}y(k-1) + \frac{5}{64}y(k-2) + \frac{3}{64}y(k-3) + \frac{1}{64}y(k-4)\end{aligned}$$

Let $y(k) = 2e^{-(k-30)^2/4} + n(k)$ where $n(k)$ is gaussian noise with $\sigma^2 = 1$

(a) Write a program to simulate the noisy signal and implement the above smoothing filter to obtain estimate $\hat{x}(k)$. Plot $x(k)$, $y(k)$ and $\hat{x}(k)$. What is the shape of the filter's impulse response function and the corresponding frequency response?

(b) Repeat (a) except instead of the estimate based on the weighted average of the adjacent signals, use ordinary averaging where

$$\hat{x}(k) = \frac{1}{9}\sum_{i=-4}^{4} y(k+i)$$

(c) Repeat (a) except use a matched filter to obtain estimate $\hat{x}(k)$.

(d) Comment on our results and state any conclusions.

9.7 This problem is meant to illustrate what it means for noise to be "white". You have learned that ideal white noise is a theoretical signal that has equal power at all frequencies – a flat spectrum for frequencies on $(-\infty: \infty)$, and a corresponding autocorrelation of $\delta(\tau)$. Of course, this signal would have infinite power. What is white noise other than a mathematical abstraction?

Another way to describe white noise is to say that each of its samples is statistically independent. For theoretical white noise, this holds no matter how fast the samples are taken. However, in digital systems, we represent signals as series of samples that have been taken at a particular rate, the sampling frequency f_s. This is the highest frequency known to the system. If a signal's power rolls off at a frequency higher than f_s, it can still be considered white for practical analysis of the system.

In this problem we will investigate the effect of sampling frequency on the power spectrum and the autocorrelation of a signal. You should see that as the signal is sampled more slowly than its correlation time – the lag between samples that are statistically dependent – it approaches the ideal of white noise.

(a) Create a random signal, then upsample it so that it has some known amount of correlation with itself. Plot the signal, its power spectral density (PSD), and its autocorrelation function (ACF), taking care to label all the axes.

(b) Now downsample your signal to a rate below your original upsampled rate. The signal should now appear to be "white," based on its PSD and ACF. Plot the signal, its PSD, and its ACF. Take care to label the axes properly.

(c) Compare the two sets of plots and explain how the same signal can result in such different spectra.

9.8 Create three independent random signals, called *A*, *B*, and *C*. Next, create the signals x, y, and z as follows:

$x = A + B;$

$y = B + C;$

$z = 5 - C;$

Which of the signals *x*, *y*, and *z*, taken pairwise, are statistically independent? Dependent? Plot the cross-correlation function magnitudes of each pair to back up your answers. Size the axes on each plot the same so that the plots are comparable.

9.9 Create a series of 100 random numbers called z_1, and make a copy of it called z_2. Next, add a new series of 50 random numbers to the beginning of z_1.

At what lag would you expect the peak of the cross-correlation between z_1 and z_2? Compute the cross-correlation and indicate where its peak(s) lie.

9.10 Clipping a signal (for example, "railing out" an amplifier) can be considered a type of noise, albeit a very predictable type of noise. In this problem, we will see that predictable and periodic noise are often worse than random noise when one is looking for a signal's frequency content.

Create an oversampled signal which is the sum of two sinusoids, as shown:

```
fs = 10; % Hz
t = [0:1/fs:500]; % sec
x = sin(2*pi*t) + 0.6*sin(0.5*2*pi*t);
```

(a) Where do you expect the peak(s) of the frequency spectrum of x? Compute and plot the power spectral density of x.

(b) Now, strongly clip this data to +/- 0.5 volts:

```
y = x; (x > 0.4) = 0.4; y(x < -0.4) = -0.4;
```

Do you expect to see the same PSD for y as for x? If different, how? Plot both x and y versus time, then plot the PSD for y. Explain the differences in the PSDs for x and y.

(c) Now modify x by adding random noise at the same amplitude level. This is called 'dithering' :

```
x_dithered = randn(1,N) + x;
```

Plot the new x versus time. Where do you expect the frequency peaks to be? Plot the PSD for the new x.

(c) Finally, strongly clip the randomized x signal as before. Plot the new y versus time as well as the PSD for the new y. Are the peaks in the correct place(s)? Why do you think this works? (Hint: look up the concept of 'dithering'.)