

ECE 220 SIGNALS & SYSTEMS I
Laboratory Project 7
Spring 2008

Assigned: April 9/10, 2008

Due: April 30/May 1, 2008

The purpose of this lab is to simulate tidal signals and to design a lowpass filter to process those signals. Your report for this project will consist of all the analytical work, Matlab plots and code, and relevant explanations.

A list of guidelines for preparing the lab report is posted on the ECE 220 website. Each student must do his or her own work on this project, however you may ask other students or any of the teaching staff for advice. As stated in the guidelines given in the ECE 220 course information packet, you should identify any students you talk to about the project.

1 Background

The rise and fall of water known as the tide is due to the the gravitational forces of the earth, moon, and sun, and the relative motion of these bodies. The tidal signal consists of a sum of sinusoidal terms with frequencies ω_n , amplitudes C_n , and phases ϕ_n . Specifically, the height of the water at time t given by:

$$s(t) = \sum_n C_n \cos(\omega_n t + \phi_n). \quad (1)$$

Note that this equation is closely related to Fourier series synthesis equation. The key difference is that the frequencies that make up the tidal signals are not harmonically related. In other words, the frequency ω_2 is not guaranteed to be an integer multiple of the frequency ω_1 .

The frequencies of the tidal components are determined by tidal physics. There are several components that have frequencies on the order of twice per day. These are known as the *semi-diurnal* components. The *diurnal* components have frequencies on the order of once per day. There are other tidal components with longer periods, such as the lunar fortnightly component which has a frequency on the order of once per two weeks. The amplitudes (C_n) and phases (ϕ_n) of these components vary with location. In other words, if you want to predict tides at Seattle, you have to know the C_n and ϕ_n for Seattle.

2 Synthesis of Tidal Signals

In this part you will write a function called `tidesynth` that synthesizes tidal signals given vectors containing the frequencies, amplitudes, and phases of each of the tidal components. Rather than using a `for` loop to sum the sinusoidal signals (as you did in Lab 1), you should implement the summation using a matrix multiply.

The function `tidesynth` should take the following arguments as inputs:

- `w`: vector of tidal frequencies (rads/day)
- `C`: vector of tidal amplitudes
- `phi`: vector of tidal phases (radians)
- `t`: vector of times (in days)

It should return the tide signal in a vector s , *i.e.*, you will call the function as follows:

$$s = \text{tidesynth}(w, C, \text{phi}, t)$$

- (a) Study how the function `ctfs_synthesis` posted on the course website works. It writes the summation for the exponential Fourier series as a matrix multiply, *i.e.*, it implements the following summation by multiplying matrices

$$\sum_{k=-N}^{+N} a_k e^{jk\omega_0 t} \quad (2)$$

Once you understand how `ctfs_synthesis` works, you should be able to create your own function to implement the summation in Equation 1.

- (b) Test your function using several signals that you choose. Explain your rationale for choosing these test signals and discuss how you verified that your function works. Your report should include plots showing the successful synthesis of these test signals.
- (c) Use your function to generate 60 days of the predicted tidal signal for the port of Seattle. Use a sampling frequency of once per hour. In other words, define $t = 0 : 1/24 : 60$. The Matlab file `seattle_tides.mat` contains the frequencies, amplitudes, and phases for the tidal components at Seattle. Include a plot in your report. You can compare to actual data for Seattle by choosing the Seattle tide station from the following webpage: http://tidesandcurrents.noaa.gov/station_retrieve.shtml?type=Tide+Data. (See link off the ECE 220 assignments page.) What are the frequencies of the tidal components contained in the Seattle data file?

3 Filtering of Tidal Signals

Suppose that you want to filter out the diurnal and semi-diurnal components of the tidal signal so that you can clearly see the semi-monthly component. You need a lowpass filter to do this. Your lowpass filter should pass components with frequencies of less than 1 cycle per week and attenuate all other signals. In this part, you will design a Butterworth lowpass filter that you can apply to the tidal signals.

Hint: you may want to check the index of the Oppenheim/Willsky book for references to Butterworth filters before beginning this section!

3.1 Poles and zeros of the Butterworth filter

An N th order Butterworth filter with real impulse response has the transfer function $H(s)$ that satisfies the following equation:

$$H(s)H(-s) = \frac{1}{1 + (s/j\omega_c)^{2N}}.$$

Recalling that the poles of the systems are the roots of the denominator, we can solve for the poles of the function $H(s)H(-s)$ as follows:

$$1 + \left(\frac{s}{j\omega_c}\right)^{2N} = 0.$$

Rearranging we obtain:

$$\left(\frac{s}{j\omega_c}\right)^{2N} = -1.$$

Note that $-1 = e^{j\pi} = e^{j(\pi+2\pi k)}$, thus

$$\left(\frac{s}{j\omega_c}\right)^{2N} = e^{+j(\pi+2\pi k)},$$

where k is an integer. (If you have trouble seeing this, draw the unit circle!) Taking the $2N$ th root of each side yields

$$\left(\frac{s}{j\omega_c}\right) = e^{+j\frac{(\pi+2\pi k)}{2N}}.$$

Moving the $j\omega_c$ term to the right-hand side and remembering that $j = e^{j\pi/2}$ yields an equation for the pole locations of the function $H(s)H(-s)$:

$$s = \omega_c e^{j\left(\frac{\pi}{2} + \frac{\pi}{2N} + \frac{\pi k}{N}\right)} \quad k = 0, 1, \dots, 2N - 1.$$

The above equation defines the locations of $2N$ poles. The poles of the causal, stable Butterworth filter defined by $H(s)$ are those in the left half plane. The poles of $H(-s)$ are those in the right half plane.

- Suppose that $N = 1$ and $\omega_c = 2$. Where is the pole of the Butterworth filter located? Suppose that $N = 2$ and $\omega_c = 1$. Where are the poles of the Butterworth filter located in this case? In general, how do the pole locations change when ω_c changes? How do the pole locations change as N is increased? For instance, consider the case when $N = 20$. Based on Equation 3.1, can you visualize where the poles of the Butterworth will be in the s -plane for this case?
- The Matlab command `butter` can be used to design analog Butterworth filters. This command can be called as follows

```
[b, a]=butter(N, wc, 's');
```

Note that the 's' is required to design an **analog** Butterworth filter. The command returns the `b` and `a` coefficients of the differential equation in the `b` and `a` vectors. You can generate the LTI system representation in Matlab using the `tf` command, *i.e.*, `sys=tf(b, a)`. Once you've done this, it is possible to plot the poles and zeros of the transfer function using the command `pzmap(sys)`.

Use this Matlab command to design Butterworth filters with several different values of N and wc (use the values you used in the previous part). Do the pole locations agree with what you computed analytically? Include plots of the poles for the different cases you considered in part a to illustrate how the pole locations depend on N and wc .

3.2 Frequency response of the Butterworth filter

The frequency response of an N th order Butterworth filter satisfies the following equation:

$$|H(j\omega)|^2 = \frac{1}{1 + (\omega/\omega_c)^{2N}}.$$

- (a) Generate several Butterworth filters with different N and ω_c values. Plot the frequency response magnitudes of these filters using the `bode` command. Verify that you obtain a lowpass filter characteristic. What effect do the parameters N and ω_c have on the frequency response? Does one of these parameters control the *rolloff* of the filter, *i.e.*, the rate at which the response decays as a function of frequency? Which parameter controls the rolloff? Using your knowledge of Bode plots, can you predict the rate of decay (*e.g.*, -20 dB/decade, -40 dB/decade, *etc.*)?
- (b) Read Section 6.4 in the Oppenheim/Willsky textbook on practical filter specifications. As noted in this section, the filter we typically constrain the gain in the passband to be greater than G_p , in other words

$$G_p \leq |H(j\omega)| \leq 1.0 \quad \text{for } |\omega| \leq \omega_p.$$

We constrain the gain the stopband to be less than G_s :

$$|H(j\omega)| \leq G_s \quad \text{for } |\omega| \geq \omega_s.$$

Using the equation for the frequency response and the definitions of G_p , ω_p , G_s , and ω_s , it is possible to derive a set of equations that determine the values of N and ω_c required to meet the filter specifications. (Note that N always has to be an integer – it is the number of poles in the system!). The Matlab command `buttord` implements these equations and solves for N and ω_c . The inputs to the `buttord` command are

- ω_p : the passband edge frequency (in radians)
- ω_s : the stopband edge frequency (in radians)
- R_p : the passband ripple defined as $-20 \log_{10}(G_p)$
- R_s : the stopband ripple defined as $-20 \log_{10}(G_s)$

Using this command experiment with different specifications. As you make ω_p and ω_s closer together, what happens to the values of N and ω_c ? Discuss your results. Note that in general we want to keep N as low as possible because lower order filters are easier and cheaper to implement.

- (c) Design a Butterworth filter that will pass all signals with frequencies of less than 1 cycle/week (1/7 cycle per day) with a minimum gain of 0.99. The filter should attenuate all signals with frequencies greater than 0.9 cycles per day. The gain in the stopband should be less than 0.0001. Determine the values of ω_p , ω_s , R_p , and R_s to send to the `buttord` command. (**Important:** don't forget to multiply frequencies by 2π to convert from cycles/day to radians/day!) Once you determine N and ω_c , use `butter` to design the filter. Plot the magnitude and phase of the filter you design for the tidal signals using the `bode` command. Does the filter meet your specifications? *Note: you may want to plot the frequency response as a function of $\omega/(2\pi)$ so that you can see where the cutoff frequency is in cycles/day instead of rad/day.* Describe your filter specifications in your report.

3.3 Application of the Butterworth filter

Use the Butterworth filter you designed in the previous section to filter the tidal signal you generated from the data contained in `seattle_tides.mat`. You can use the `lsim` command to implement the filtering. Plot the resulting filtered signal. Does it contain only the lowest frequency component of the tide?

Since your filter should remove all but the lowest frequency component of the tide signal, the only thing left should be a single cosine wave. Since you know the amplitude, phase, and frequency of this component, you can predict what it looks like at the input to the filter. (You do this by running your synthesis code with only that component.) Plot this component on the same graph as your filtered signal. Are the signals the same? If they differ, explain how you could have predicted these differences.

Hint: you know what happens to cosine signals when they pass through LTI systems. Use this knowledge to predict the output of the filter!