ECE 251 Lab 1 Explanation

Stephen Kemp

1. The maximum PSD of the wideband thermal noise (Johnson noise) introduced by the receiver is found as N0 /2 = kT/2. Where k = 1.381 J/K (Boltzmann constant) and T = 300K in this case. The variance of a time-domain random process X is defined as Var(X) = E[|X-u|2] where u is the mean of X. In our case, the mean of the white noise is u = 0, so we’re left with Var(X(w)) = E(|X(w)|2). The PSD of a random process is defined as limN->inf 1/N E{X(w) X(w)\*} which -> PSD = E(|X(w)|2). So the variance of the white noise is equal to the PSD of the process which is N0/2. Now this is true for a real white-noise signal. The problem asks to find the variance sig2 of a complex white-noise signal. The expected value of a complex random variable is defined as E[Z] = E[R(Z)] + jE[I(Z)], so the variance of the same random variable will be E[Z2] = E[R(Z)2] + jE[I(Z)2]. So sig2 = sig2/2 + sig2/2. The variance of the real and imaginary white noise will be sig2/2 = N0/4.
2. For the filter design, we want a cutoff frequency of 10kHz, with a passband gain of 20dB and a stopband attenuation of -50dB. I have chosen a sampling frequency of 40kHz so we can get a signal bandwidth of 20kHz. This will conveniently put the 10kHz cutoff in the center of the plot and will give plenty of room to view the stopband starting at 12kHz. I have chosen a filter order of 35 so that the stopband starts at approximately 12kHz. The stopband gain is currently -100dB, and I’m not sure how to modify that to -50dB using fir1 or scaling.
3. To generate the complex white noise, I used the randn function which generates standard normally distributed values. I then multiplied the resulting vector by sqrt(variance) to scale the standard deviation of the normal distribution, because the default std dev is 1 (normal). I generated one vector for the real noise and one for the imaginary noise and created a complex valued vector using the other two. For the filter function, the vector of fir filter coefficients become the numerator coefficients of the transfer function, and since the filter is fir, the denominator coefficient is just 1.
4. To calculate the PSD, I created a function which takes the input data/time vectors and the desired number of chunks, and outputs the PSD vector. It does this using the Wiener-Kinchin method outlined in the assignment notes. To achieve the required resolution of 10Hz, I used the require 50 chunks combined with a 5-second-long noise sample.