Problem 12.1

(a) Consider the standard approximation to the Gaussian process $x(t)$. What is its autocorrelation function $R_x(\tau)$? Is $x(t)$ wide-sense stationary?

(b) Now, generate a discrete version of $x(t)$ in MATLAB and plot the estimate for discrete $R_x(\tau)$:

\[
N = 500;
x = \text{randn}(N, 1);
Rx = \text{xcorr}(x)/N;
\text{stem}(-(N-1):(N-1), Rx);
\]

Zoom into the area near 0 on the x-axis with the command: `axis([-5 5 -.1 2])`. Does this match with your calculations? You should also be able to verify that $R_x(0) = E[(x(t))^2]$ (in MATLAB, $R_x(0)$ is stored in $Rx(500)$). This is the power or variance of the stationary process.

(c) The power-spectral-density (PSD) $S_x(f)$ is the Fourier transform of $R_x(\tau)$:

\[
S_x(f) = \int_{-\infty}^{\infty} R_x(\tau) e^{-j2\pi f \tau} d\tau.
\] (1)

Calculate what that is. We can plot the discrete version from our estimate of $R_x(\tau)$ in MATLAB:

\[
Sx = \text{abs} (\text{fft}(Rx));
\text{plot}((-\text{N-1}):\text{N-1})/(2*\text{N}), Sx);
\]

When finding the Fourier transform in MATLAB, we use the `fft` command which computes the Discrete Fourier Transform (DFT) using the Fast Fourier Transform (FFT) algorithm. The frequency axis for the DFT will be normalized to be between $-\frac{1}{2}$ and $\frac{1}{2}$ when we plot it. (Do you remember why this is, from EE3015?)

Another way to estimate $S_x(f)$ is to compute $\frac{1}{N}|X(f)|^2$, where $X(f)$ is the Fourier transform of $x(t)$. In MATLAB, the discrete version is:

\[
Sx\_est = \text{abs} (\text{fft}(x)) \cdot 2/N;
\]

This method is usually preferred for estimating the PSD. The PSD is actually symmetric about 0 in this case, so we can simply plot the “right half” of the PDF from 0 to $\frac{1}{2}$:

\[
\text{plot}(0:N/2-1)/N, Sx\_est(1:N/2));
\]

Problem 12.2

Now, we will try a slightly different process. Since we are working in MATLAB, we can simply focus on the discrete version. Consider $n = 2, 3, 4, \ldots$:

\[
w(n) = \frac{1}{2}x(n-1) + \frac{1}{2}x(n),
\]

and assume $w(1) = x(1)$. This is a “moving average” process, because it takes the average of the current and previous value. Generate $w(n)$ in MATLAB as follows.
\texttt{w(1) = x(1);}
\texttt{for i=2:N}
\texttt{ \hspace{1em} w(i) = 0.5*x(i-1) + 0.5*x(i);}
\texttt{end}

Now plot \(x(n)\) and \(w(n)\) for \(n = 1, 2, \ldots, 50\):
\texttt{plot(1:50,x(1:50),1:50,w(1:50))}

You should see that \(x(n)\) and \(w(n)\) are very similar, but \(w(n)\) is “smooth” whenever \(x(n)\) has a sharp change. For this reason, \(w(n)\) is said to be the output of a “smoothening” filter, or a low-pass filter (LPF).

(a) Since \(w(n)\) can be thought of as a “filtered” version of \(x(n)\), we can write it as the convolution with an impulse response \(h(n)\):
\[
 w(n) = \sum_{i=-\infty}^{\infty} h(i)x(n-i).
\]
Notice here that \(h(n)\) is not random. From the definition of \(w(n)\), find the impulse response \(h(n)\), and its Fourier transform \(H(\phi)\), defined as
\[
 H(\phi) = \sum_{i=-\infty}^{\infty} h(n)e^{-j2\pi\phi n}
\]
This is the Discrete Time Fourier Transform (DTFT), and compared to (1), it is different because we have replaced the integral with a sum (because \(h(n)\) is discrete), and changed the frequency variable to \(\phi\) instead of \(f\), to avoid confusion with the continuous version. You may recall that \(H(\phi)\) is called the frequency response of the filter.

(b) Recall that convolution in the time domain is the same as multiplication in the frequency domain, so that \(W(\phi) = H(\phi)X(\phi)\). This is similar to what you already learned in EE3015, except that the input \(x(n)\) and the output \(w(n)\) are random signals, rather than deterministic ones. This means you cannot really say what the exact values of \(x(n)\) and \(w(n)\) will be, but you can talk about their statistics and properties. One such property is that:
\[
 S_w(\phi) = |H(\phi)|^2 S_x(\phi).
\]
Calculate the power spectral density \(S_w(\phi)\) (again, we are using the variable \(\phi\) instead of \(f\) following the convention used in the textbook, but it really is the same thing).

(c) We will estimate the discrete version in MATLAB:
\begin{verbatim}
Sw_est = abs(fft(w)).^2/N;
plot( 0:N/2-1)/N, Sw_est(1:N/2));
\end{verbatim}

Does this match with your calculations? What is the main difference between the shapes of \(S_x(\phi)\) and \(S_w(\phi)\)? Does this make sense, since \(h(n)\) is a low-pass filter?

(d) The discrete autocorrelation function \(R_w(n)\) is the inverse Fourier Transform of \(S_w(\phi)\). Compute what that is. You can verify your answer with the MATLAB estimate:
\begin{verbatim}
Rw = xcorr(w)/N;
stem(-(N-1):(N-1), Rw);
axis([-5 5 -.1 2]);
\end{verbatim}