ECE310 – Project 5

Jonathan Lam

December 20, 2020

1 Setup

(Denote the correlation of signal s by $\phi_s(\tau)$, and the PSD of a signal by $\Phi_s(\omega)$.)

We have a random Gaussian signal x s.t. $\Phi(x)$ has $\sigma^2 = 1$. We also have a channel with some frequency response $|H(e^{j\omega})|^2$. If the result of x passed through the filter h is y, then:

$$\Phi_y(e^{j\omega}) = |H(e^{j\omega})^2|\Phi_x(e^{j\omega}) = |H(e^{j\omega})|$$

Thus estimating $\Phi_y(e^{j\omega})$ is equivalent to estimating $|H(e^{j\omega})|^2$. We are given a 512-point signal $y[n]$ and use this to estimate $|H(e^{j\omega})|^2$ by estimating its PSD. We are also given a 512-point sample of $|H(e^{j\omega})|^2$ to check our estimates by the error criterion (1). These signals are displayed in Figure 1.

$$\epsilon = \frac{1}{N}\sum_{k=0}^{N-1}\left|\left|\hat{H}\left(e^{j\omega}\right)\right|^2 - \left|H\left(e^{j\omega}\right)\right|^2\right|^2 \tag{1}$$

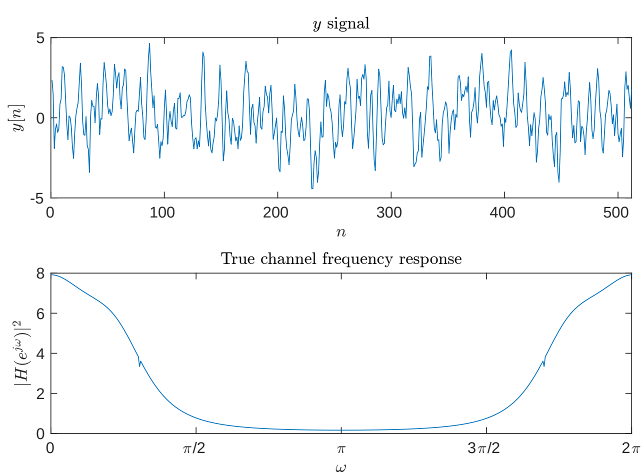


Figure 1: The provided signals $y[n]$ and $|H(e^{j\omega}|^2$

2 Autocorrelation and MATLAB (§A)

We can take the cross-correlation of x with itself (i.e., the autocorrelation) using the MATLAB function `xcorr`, which is similar to taking the convolution with the reverse of x (i.e., taking inner products at each shift).

```
y1 = y(1:32);
autocorr1 = xcorr(y1, y1, 'biased');
autocorr2 = conv(y1, flip(y1));
```

The difference is that the `xcorr` function normalizes to a biased estimate with $N = 32$, while the convolution does not.

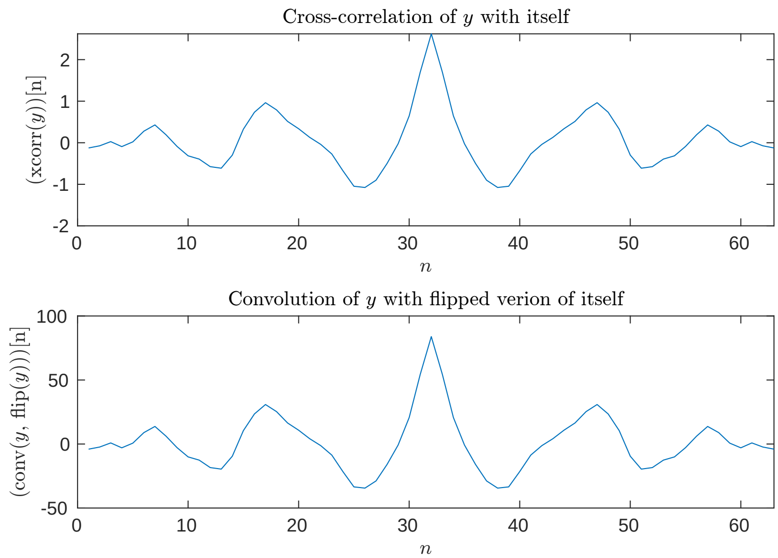


Figure 2: Comparison of `xcorr` vs `conv`

2.1 Biased vs. unbiased estimates (§A.1)

The term "biased" in this context means that the correlation is equal to the convolution scaled down by a factor of N, while an unbiased estimate is scaled down by a factor of $N - |n|$. The latter produces the expected value in the limit as $N \to \infty$ (hence "unbiased"), but suffers from large variance when n approaches N when a finite number of samples are used. The biased estimate does not approach the expected value in the limit, but it does not suffer from the large variance problem.

2.2 Deterministic autocorrelation (§A.2)

We can calculate the (deterministic) autocorrelation using:

$$\phi_s[m] = \frac{1}{N} \sum_{n=0}^{N-1} s[n+m]s[n]$$

A MATLAB implementation for this to calculate the autocorrelation of y_1 is:

```
phi_y1 = zeros(1, 32);
for i = 1:N_y1
    phi_y1(i) = sum(y1(1:N_y1-i+1) .* y1(i:N_y1)) / N_y1;
end
phi_y1 = [flip(phi_y1) phi_y1(2:end)];
```

This gives us exactly the same result as calculated by `xcorr` (which confirms that our result is correct).

The Fourier transform of this autocorrelation function is a positive, real function because it is the PSD (by Wiener-Khinchin), of which each point is positive and real (the power of a given frequency). However, when we plot the magnitude and phase, we do not get a positive real function – this is plotted in Figure 3. (This probably due to some symmetry of the problem.) (We do get linear phase, however – not sure if this is relevant.)

2.3 Plotting different PSD estimates (§A.3)

Three estimates of the PSD are plotted in Figure 4:

1. `abs(fft(xcorr(y1, 'biased'), 64))`
2. `abs(fft(y(1:32), 64)).^2`
3. `abs(fft(y(1:64), 64)).^2`

The first method uses Wiener-Khinchin to calculate the PSD by taking the FFT of the autocorrelation function. The second and third methods calculate the PSD directly by taking the power (magnitude squared) of the FFT. The third uses more samples of y to calculate the FFT.

It seems that the second method is less noisy than the first method. The third method, despite taking more samples of y, is more noisy than the second method; this is the strange property that is detailed in section 10.5.2 of the textbook, so taking fewer samples for the periodogram is actually better (less noisy).

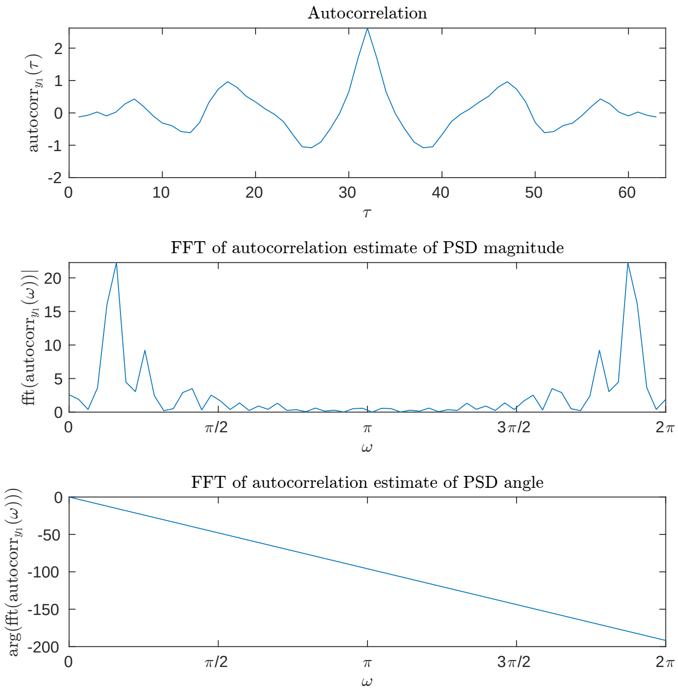


Figure 3: Autocorrelation and its FFT (estimate of PSD)

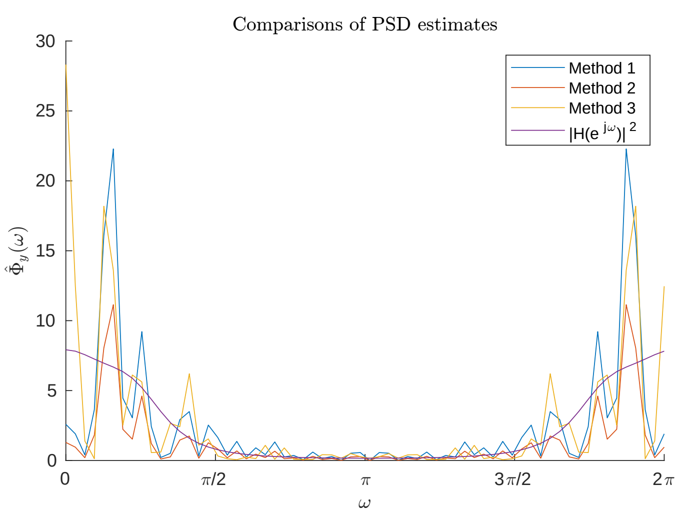


Figure 4: Comparison of the three PSD estimation methods described in §A.3

3 Nonparametric PSD estimation (§B)

3.1 PSD estimates with different sample and FFT lengths (§B.1-2)

The periodogram is calculated similar to the second method above (taking the FFT, and then finding its magnitude squared). We compare the result when 32 samples are taken (i.e., y_1) and a 64-point DFT is performed (same as method 2 in §A.3), and when all 512 samples are taken and a 1024-point DFT is performed. The results are shown in Figure 5. The same property is displayed as in Figure 4: more samples causes a noisier estimate of the PSD. These two methods have errors of 7.5039 and 7.5488 using the error measure (1).

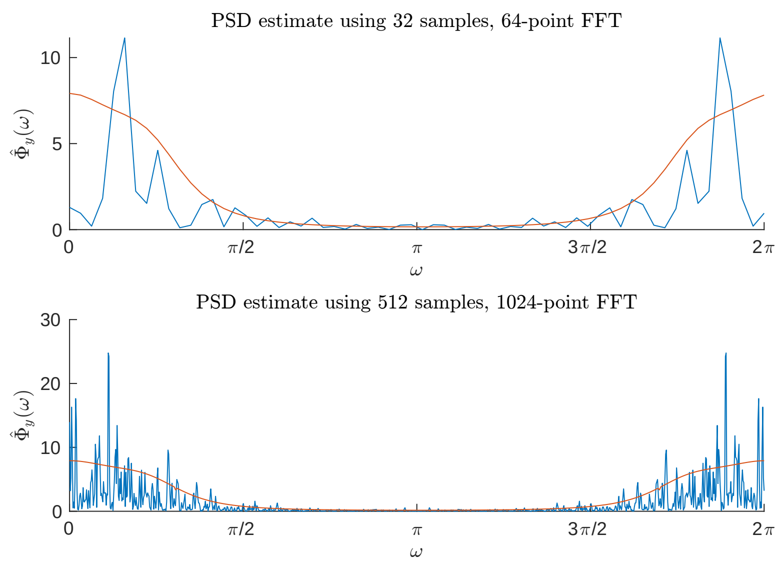


Figure 5: Comparison of PSD estimation methods with different numbers of samples and FFT sizes

3.2 Improving results by averaging (§B.3)

We can try to improve the results by taking multiple, shorter periodograms and averaging the results. Here, we break up the signal into 32-sample chunks, take the 64-point periodogram of each chunk, and average the results. This provides a smoother estimate and a lower error of 3.1403. The results are displayed in Figure 6.

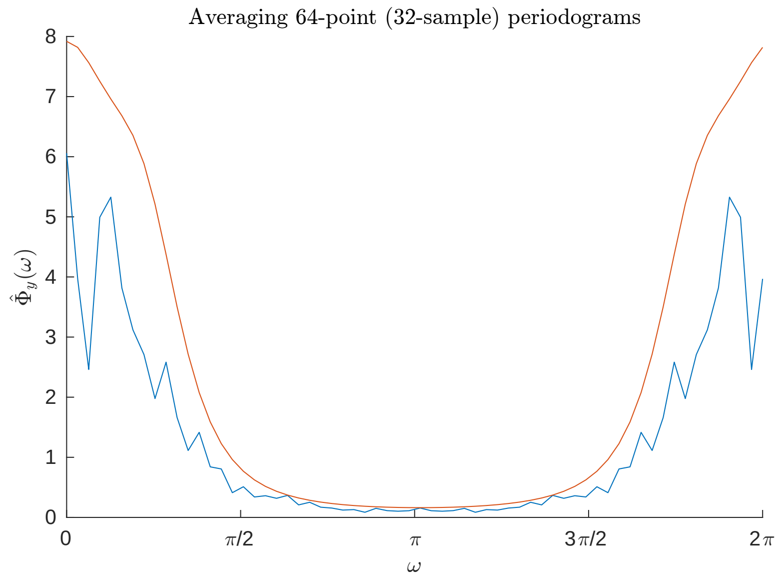
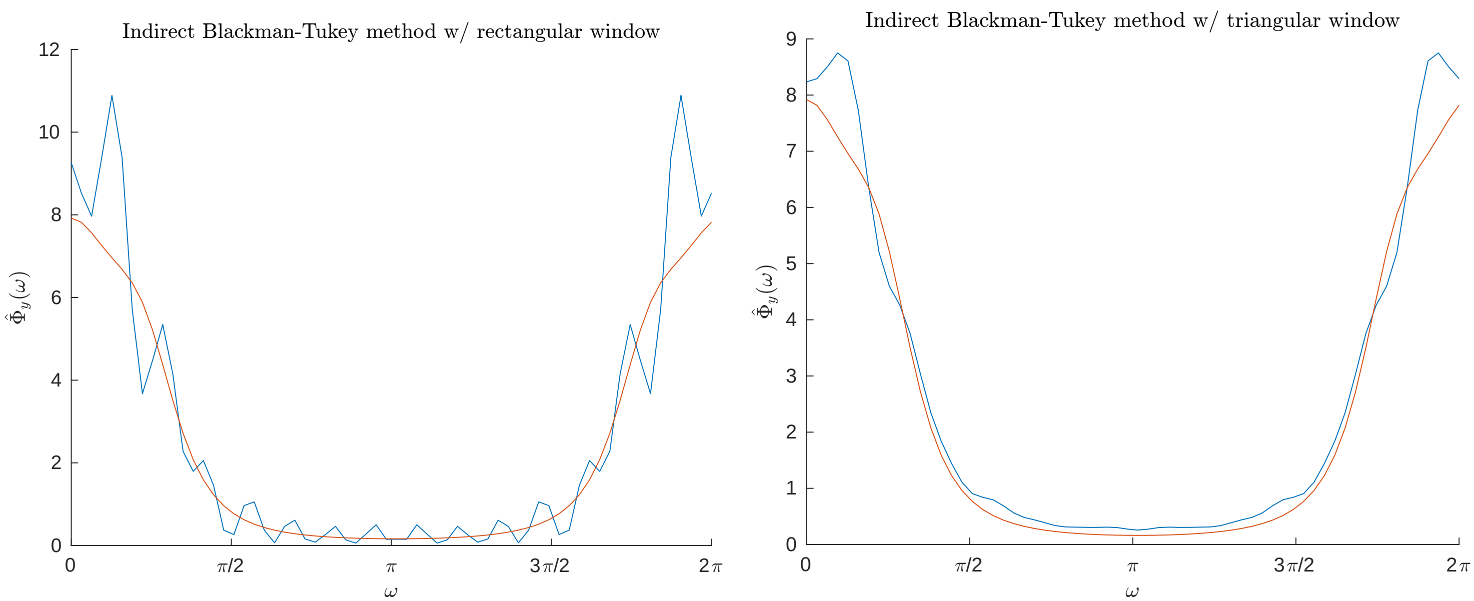


Figure 6: Averaging short periodograms produces a (relatively) smoother and more accurate result

3.3 Indirect Blackman-Tukey method (§B.4)

The Blackman-Tukey method is the first method discussed (of taking the FFT of the autocorrelation). Now, we take the autocorrelation of the entire signal ($N = 512$), then multiply this by a small rectangular centered at zero (all samples $|m| \leq 15$). Then we take the 64-point FFT. We see that this is somewhat noisy, with some "ripple" near $\omega = \pi$, but the tail behaviors are much closer to the actual function. This is most likely due to the fact that we used a biased estimator, so that the variance near the ends of the FFT is much lower. The result is plotted in Figure 7a, and the error is 1.1899.

We are also asked to do the same procedure, but windowing the autocorrelation with a 31-point triangular window (rather than a rectangular window) centered at $\tau = 0$. This generates the result shown in Figure 7b. This is much smoother and much closer on average to the true PSD than all of the other methods. Multiplying by a triangular window is similar to convolution by its Fourier transform, which has a smoothing effect; the tradeoff for losing variance is that we lose frequency resolution of the PSD, but the latter is not a concern in this case.



(a) Rectangular window

(b) Triangular window

Figure 7: PSD estimate using Blackman-Tukey method with two different symmetric windows of length 31 (centered at $\tau = 0$)

3.4 Methods summary and B-T windowing (§B.5)

The summary of the methods is shown in Table 1.

Method	Error
64-point periodogram using first 32 samples (§B.1)	7.5039
1024-point periodogram using all 512 samples (§B.2)	7.5488
Averaged 64-point periodograms (32 samples each) (§B.3)	3.1403
Blackman-Tukey w/ length-31 rectangular window (§B.4)	1.1899
Blackman-Tukey w/ length-31 triangular window (§B.4)	0.2768

Table 1: Summary of methods and errors

The Blackman-Tukey method performed better than the periodogram method here – this is probably due to the fact that it is smoother than the periodograms. The Blackman-Tukey method using the triangular window performed by far the best – even though we lost frequency resolution, the expected shape was smooth so this wasn't a problem here.

The periodogram method that performed best was the averaging of small periodograms. The 64-point periodogram was not very accurate, and the 1024-point periodogram was very noisy; combining the better parts of both (less noise and smaller bias) in the average made it perform the best.

The shape of the averaged periodogram was pretty similar to that of the Blackman-Tukey with the rectangular window, but it (as well as the other periodogram estimates) all seem systematically lower than the real PSD. I'm not sure what the reasoning behind this is, but multiplying by a factor of roughly 1.7 seems to lower all of the error estimates; for example, multiplying the averaged periodogram by 1.7 (Figure 8) lowered the error to 0.9070, which is lower than for Blackman-Tukey with the rectangular window. This is for future analysis.

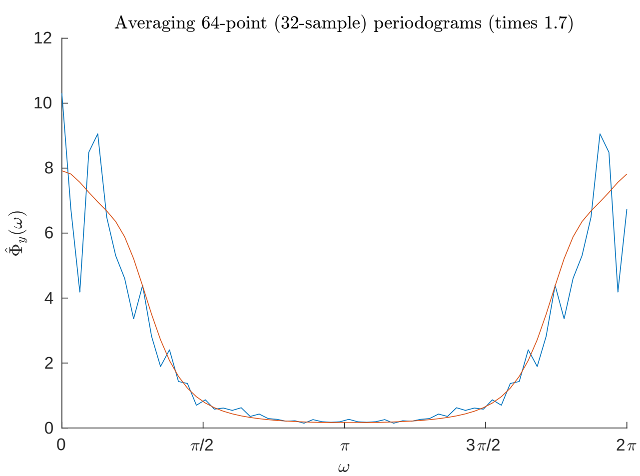


Figure 8: Multiplying averaged periodograms by a scalar factor lowers the error