Report:

Teresa Algarra Ulierte
Student ID: teral436
Personnummer: 970628T129

1 Study 1: Modelling Signals

1.1 Theoretical Background:

First of all, we will say that a stochastic process is defined as random numerical values changing over time, that is, a function of stochastic variables depending on time. These processes are used as the mathematical model to explain random phenomena, such as the noise that can appear in signals in the field of telecommunications. This noise can be characterized by its *Autocorrelation Function* ACF and its *Power Expectral Density* PSD.

The *Autocorrelation Function* or ACF shows the correlation between two samples of the same process separated by a certain time. We can represent it as:

$$r_x[k] = E\{X[n+k]X[n]\} \tag{1}$$

The *Power Expectral Density* or PSD tells us how the energy of the signal distributes depending on the frequency. It can be calculated as the Fourier Transform of the ACF:

$$R_X[\theta] = \mathcal{F}\{r_x[k]\} \tag{2}$$

If we have a LTI filter with frequency response $|H(\theta)|$ and a WSS input $X[n]$, then the PSD of the output $Y[n]$ is:

$$R_y(\theta) = R_X(\theta)|H(\theta)|^2 \tag{3}$$

Given this information we will analyze the task of this study. We will be using White Gaussian Noise, which is a particular case of stochastic process that has mean $m_x = 0$ and constant PSD $R_X[\theta] = R_0$. We will need to create two LTI filters:

- A high-degree low-pass filter approximated by an ideal filter
- A simple low-degree low-pass filter

We will put the noise through those two filters and characterize the results with the ACF and the PSD. We will create the noise with the MatLab function *rand.* We will use $R_X = 10$ for our theoretical calculations, which means that we will multiply our random array by $sqrt10$ to achieve this value in our estimations.

1.2 Theoretical Analysis:

1.2.1 High-degree Filter:

We have to use a high-degree low-pass filter that can be approximated as an ideal filter, so we will use a rectangle to simulate this filter. Our cut-off frequency will be $f_c = 0.1$, so $\theta_0 = 0.2$.

$$H[\theta] = rect(\frac{\theta}{\theta_0}) = rect(\frac{\theta}{0.2}) \tag{4}$$

The filter is digital and, therefore, periodical, so it can be defined as:

$$H[\theta] = \begin{cases} 1 & 0 \le f < 0.1 \\ 0 & 0.1 \le f < 0.9 \\ 1 & 0.9 \le f < 1 \end{cases} \tag{5}$$

This is how our filter looks:

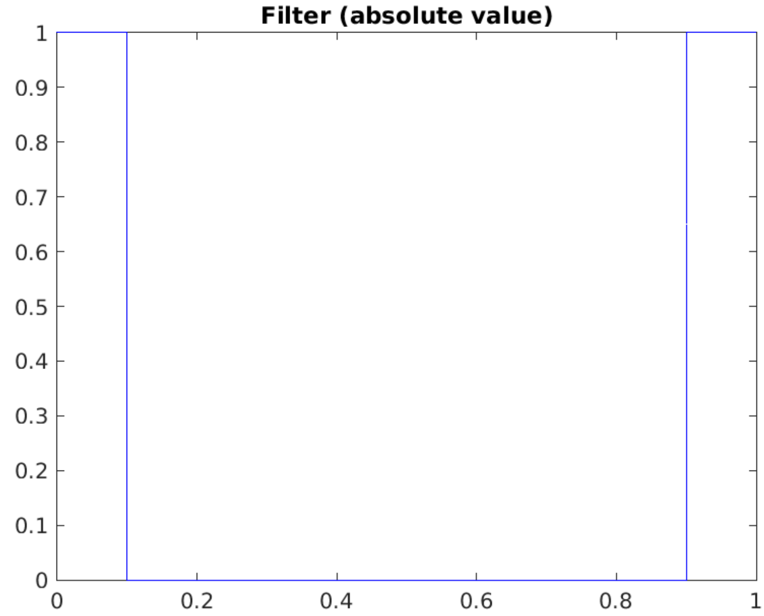

Figure 1: Ideal Filter

We will start by calculating the PSD of the filtered signal. If we use formula (3), we have:

$$R_y(\theta) = R_X(\theta)|H(\theta)|^2 = R_X|rect(\frac{\theta}{\theta_0})|^2 = R_X|rect(\frac{\theta}{0.2})|^2 = \begin{cases} 10 & 0 \le f < 0.1 \\ 0 & 0.1 \le f < 0.9 \\ 10 & 0.9 \le f < 1 \end{cases} \tag{6}$$

The resulting graph is:

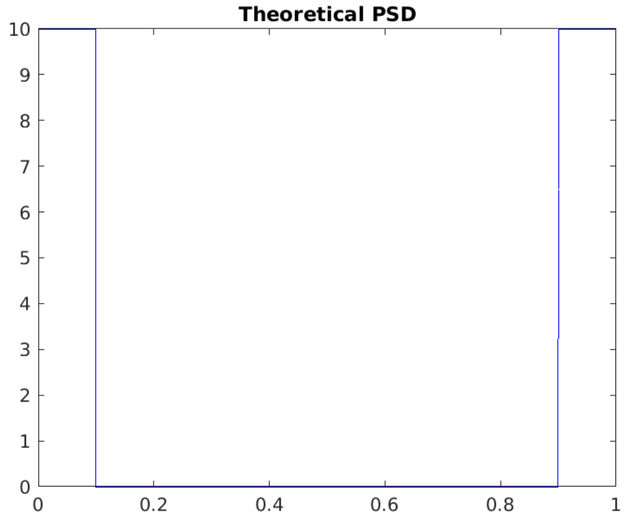

Figure 2: Theoretical PSD

To find out the ACF of the signal, we just have to calculate the inverse Fourier Transform of the PSD:

$$\begin{aligned} r_y[k] &= \mathcal{F}^{-1}\{R_X(\theta)\} = \mathcal{F}^{-1}\{R_X|rect(\frac{\theta}{\theta_0})|^2\} = \\ &= R_X \cdot \theta_0 \cdot sinc(\theta_0 k) = 10 \cdot 0.2 \cdot sinc(0.2k) = 2 \cdot sinc(0.2k) \end{aligned} \tag{7}$$

The resulting graph is:

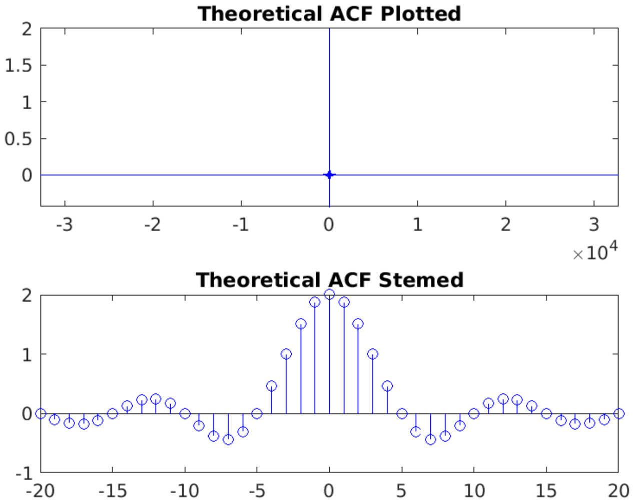

Figure 3: Theoretical ACF

1.2.2 Low-degree Filter:

For the low-degree filter, we will use the formula:

$$H[\theta] = 0.5 \cdot (1 + e^{-i2\pi\theta}) \tag{8}$$

We multiply by 0.5 so that the maximum value of the filter is 1 in stead of 2.
This results in the following graph:

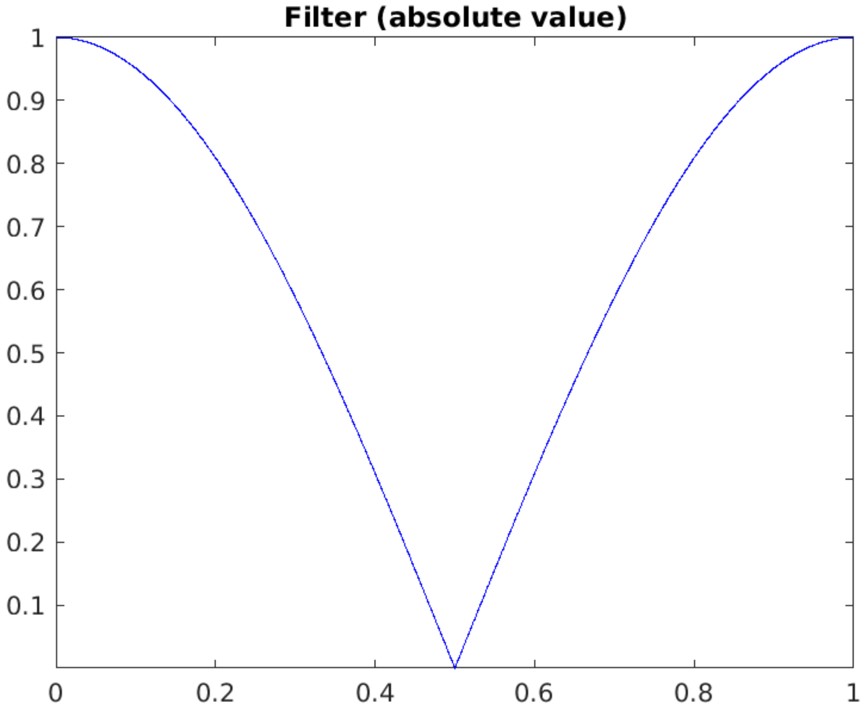

Figure 4: Low-Degree Filter

As before, we use formula (3) to get the PSD:

$$\begin{gathered}
R_y(\theta) = R_X(\theta)|H(\theta)|^2 = \\
= R_X|0.5 \cdot (1 + e^{-i2\pi\theta})|^2 = R_X \cdot 0.25|(1 + e^{-i2\pi\theta})|^2 = \\
= R_X \cdot 0.25 \cdot 2(1 + cos(2\pi\theta) = 10 \cdot 0.25 \cdot 2(1 + cos(2\pi\theta) = \\
= 5(1 + cos(2\pi\theta))
\end{gathered} \tag{9}$$

The graph that formula gives us is:

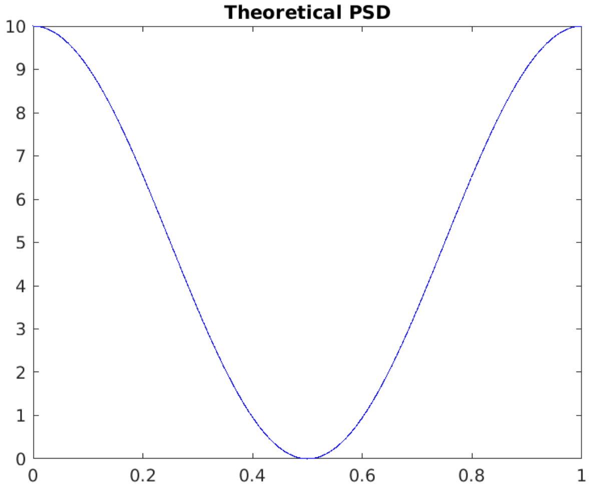

Figure 5: Theoretical PSD

Again, to calculate the ACF of the signal, we do the inverse Fourier Transform of the PSD:

$$\begin{gathered} r_y[k] = \mathcal{F}^{-1}\{R_X(\theta)\} = \mathcal{F}^{-1}\{5(1+cos(2\pi\theta)\} = \\ = 5\delta[k] + 2.5\delta[k-1] + 2.5\delta[k+1] = \begin{cases} 2.5 & k=-1 \\ 5 & k=0 \\ 2.5 & k=1 \end{cases} \end{gathered} \tag{10}$$

Therefore, our graph is:

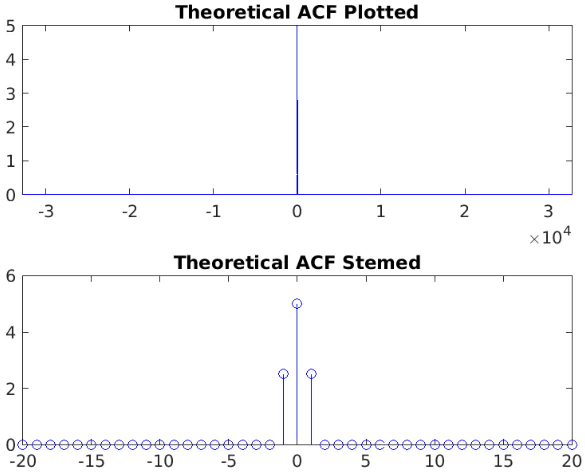

Figure 6: Theoretical ACF

1.3 Estimations:

Let's move on to the estimations. For the ACF we take the Bartlett's ACF estimate:

$$\hat{r}_y[k] = \begin{cases} \sum_{n=0}^{N-|k|-1} x[n+|k|]x[n] & 0 \leq n < N \\ 0 & \text{elsewhere} \end{cases} \tag{11}$$

For the PSD we can just do the Fourier Transform MatLab function *fft(x)*.

1.3.1 High-degree Filter:

For the ideal filter, the estimation for the ACF is the following:

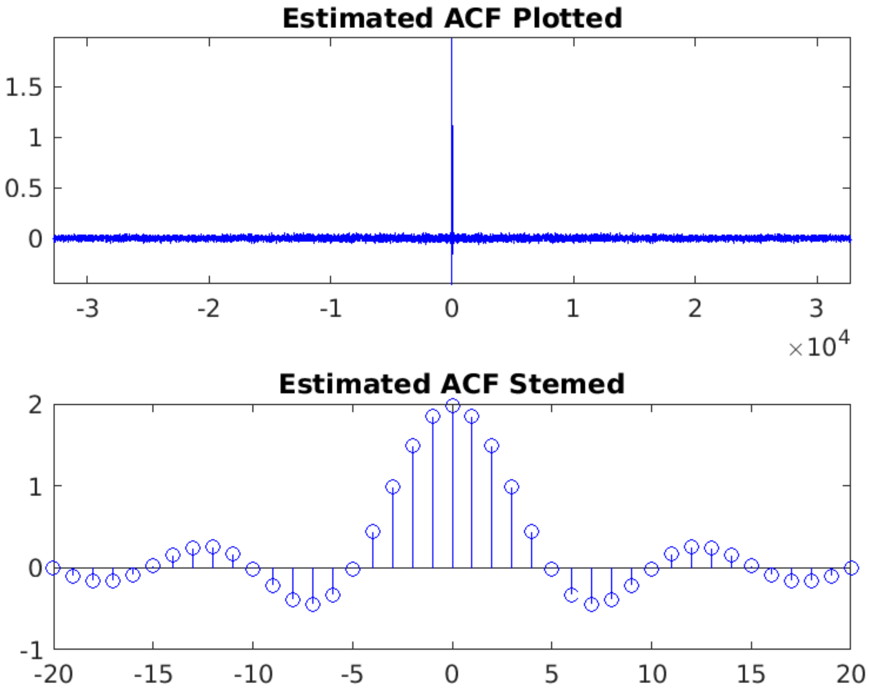

Figure 7: Estimated ACF

As we can see, it is very close to the the theoretical estimation. The plotted graph is slightly more noisy, but the stemed graph is practically the same.

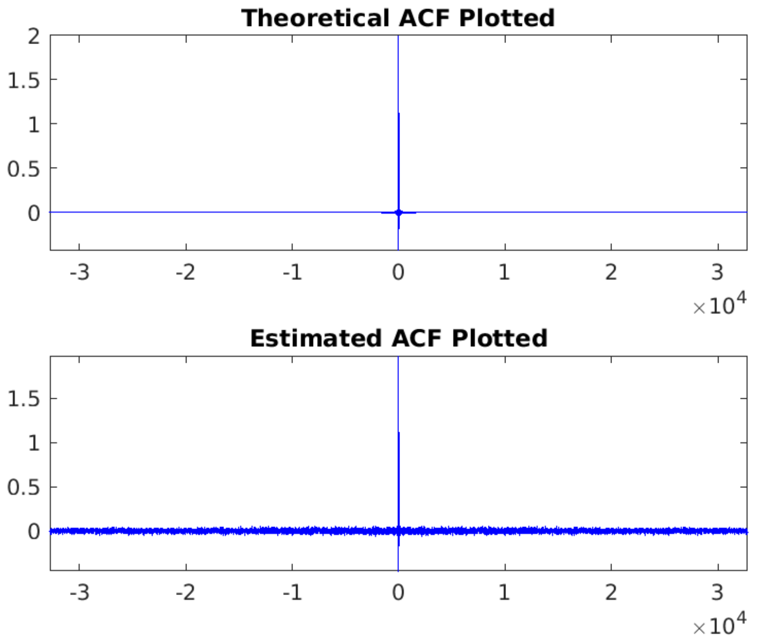

Figure 8: Compared ACF

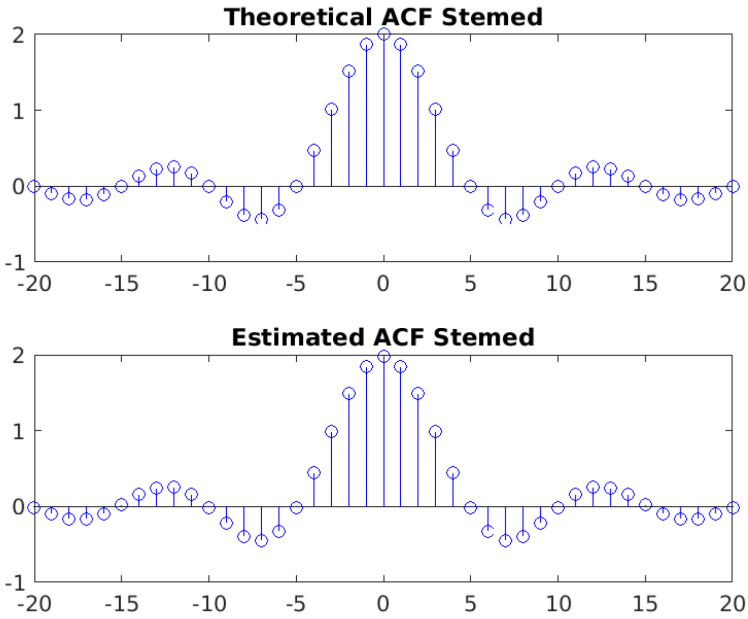

Figure 9: Compared ACF

The PSD graph resulting from the Fourier Transform of the above is:

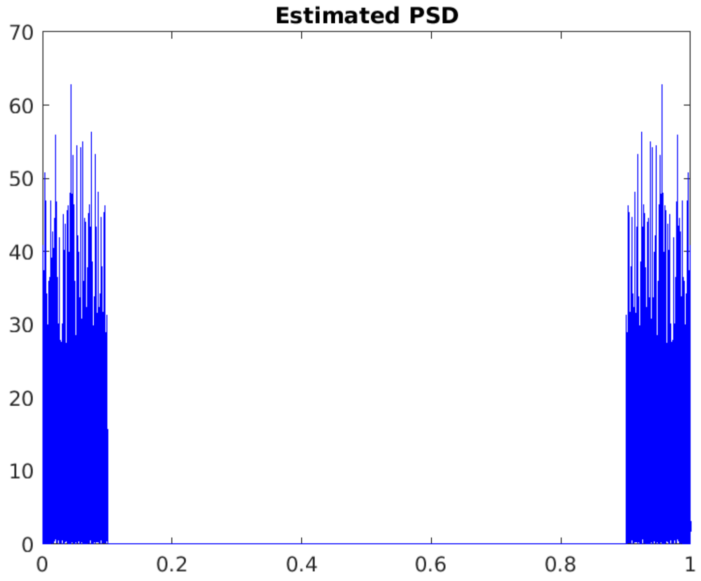

Figure 10: Estimated PSD

On plain sight, it looks very different from the theoretical PSD graph, but because it has a lot of noise. During study 2 we will see that the values once we average or smooth the PSD are the same as the theoretical analysis.

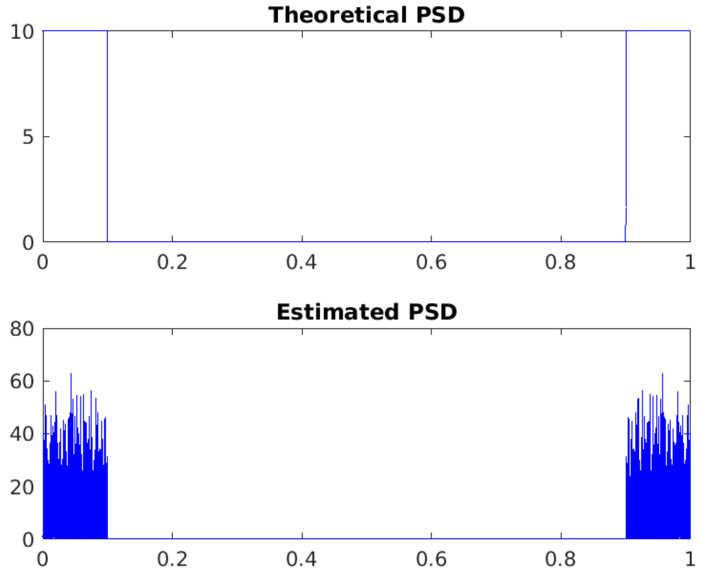

Figure 11: Compared PSD

1.3.2 Low-degree Filter:

For the simple filter, the estimation for the ACF results in this graph:

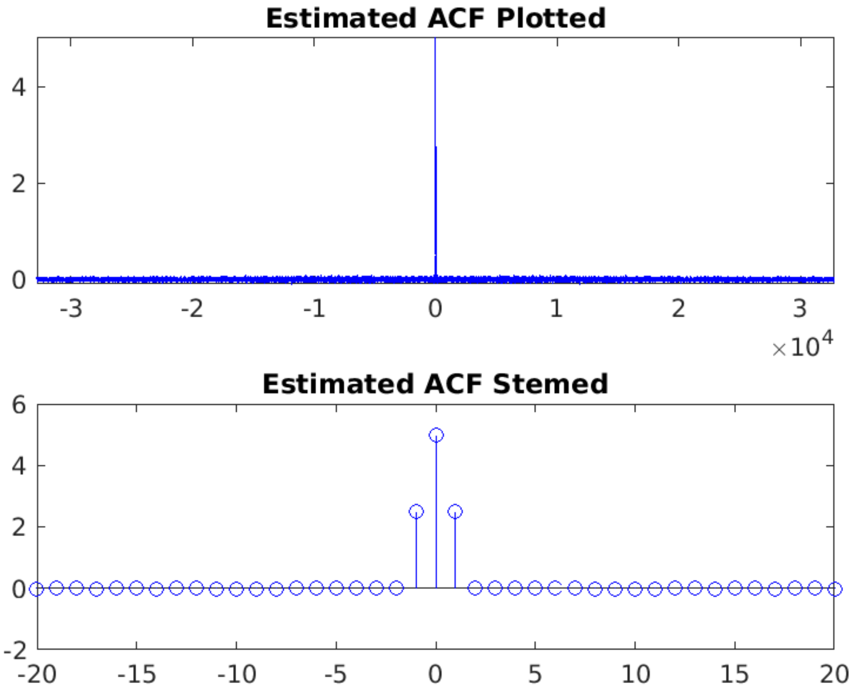

Figure 12: Estimated ACF

Again, it is very close to the the theoretical estimation. Once more, we can see that the plotted graph is ticker in the x-axis, and the stemmed graph has values slightly under 0, which we didn't have in the theoretical analysis, but it is practicaly imperceptible.

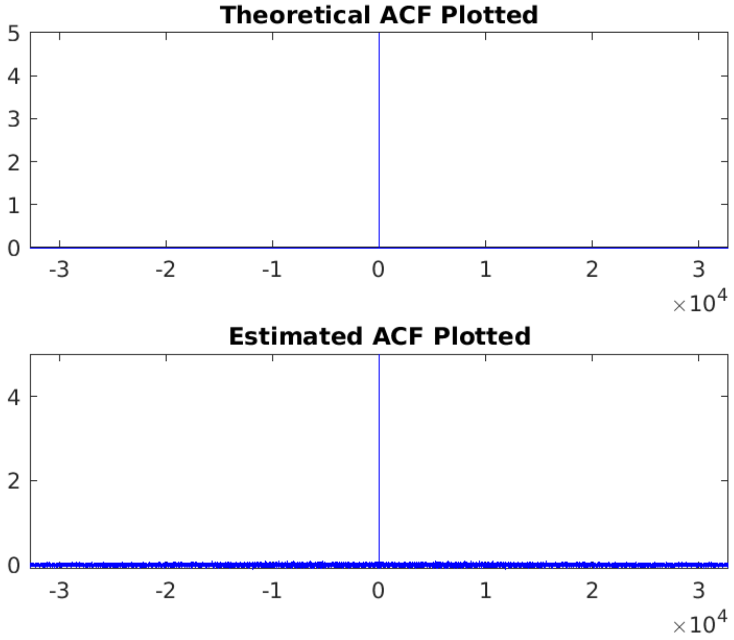

Figure 13: Compared ACF

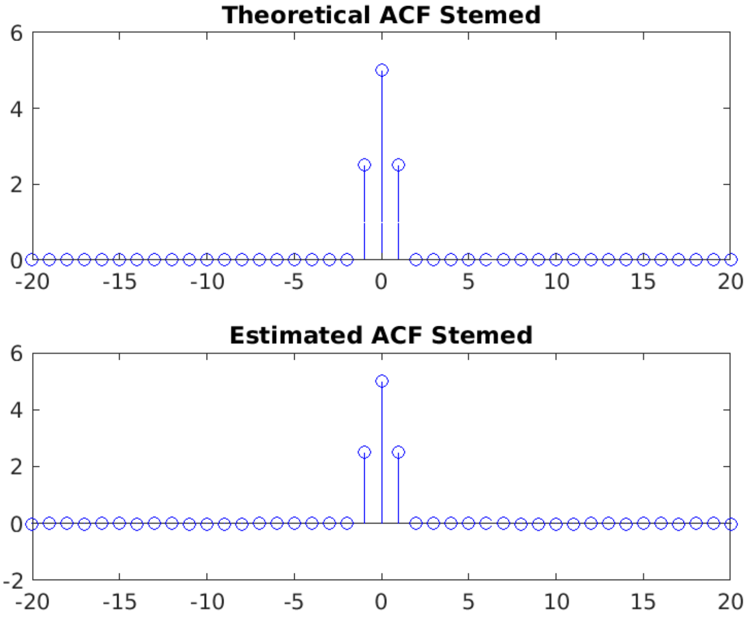

Figure 14: Compared ACF

The PSD graph that we get from the Fourier Transform of the ACF is:

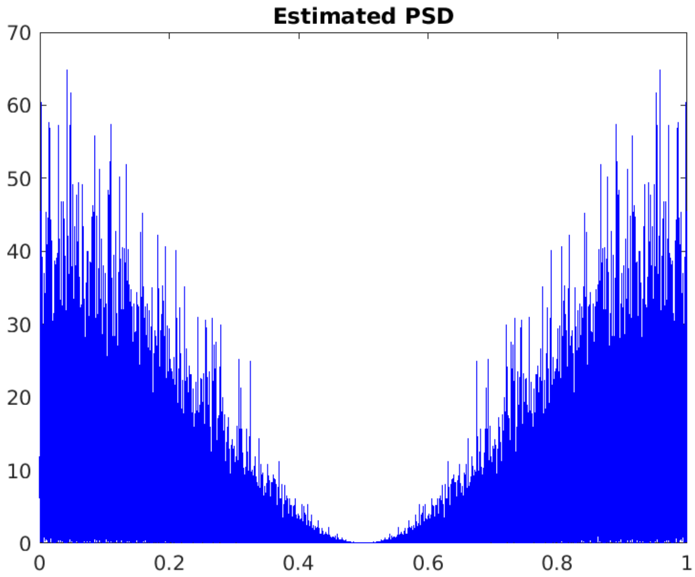

Figure 15: Estimated PSD

Here we have the same case as before, now we have a pointy and noisy draft of the PSD that, in the next study, will start to resemble more to the theoretical PSD.

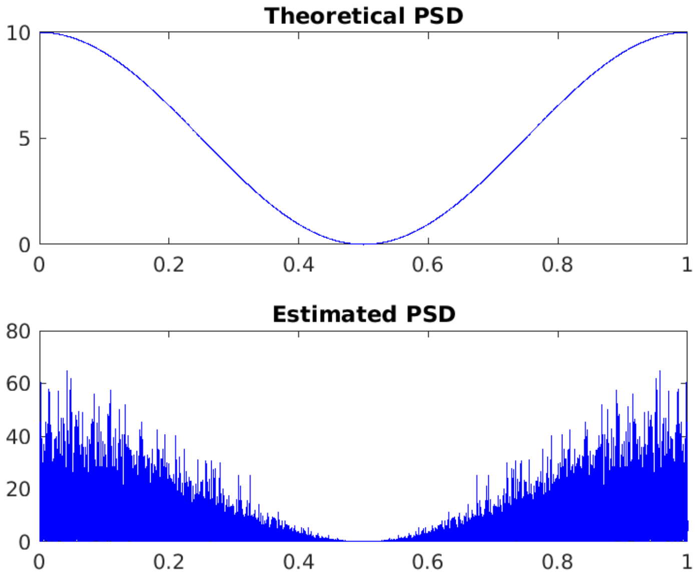

Figure 16: Compared PSD

2 Study 2:

2.1 Theoretical Background:

In this second study, the aim is to improve the estimates done in the first study. We will use all the graphs we got from study 1, and improve them in two different ways:

- **Windows:** For the windowing technique, we will take five different windows:
  - Rectangular window
  - Triangle window
  - Hamming window
  - Bartlett window
  - Blackman-Harris window

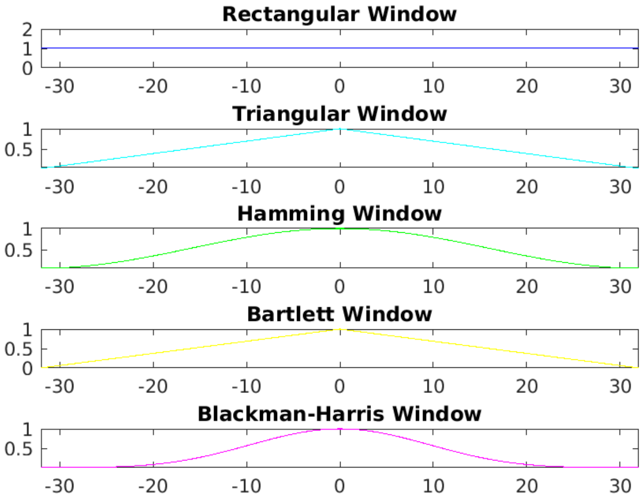

Figure 17: Compared PSD

  We will pad them with zeros and then use them as if they were filters to smooth our graphs. We will apply them to the ACF and then we will get the PSD with the Fourier Transform again.

- **Averaging:** The basis of this technique is spliting the signal is several parts, estimating each one of them and then averaging all the results. This allowes us to reduce the variance and the pointy look of the PSD. We follow the following formula:

$$R_{X_{Av}}[\theta] = \frac{1}{K}\sum_{k=1}^{K} \hat{R}_{X,k}[\theta] \tag{12}$$

  We will use three different values to compare them, 2^4, 2^6 and 2^8. Once again, we will apply it to the ACF and then switch to frequency domain.

2.2 Smoothing:

We will start with the window smoothing.

2.2.1 High-degree Filter:

Here we have the ACF:

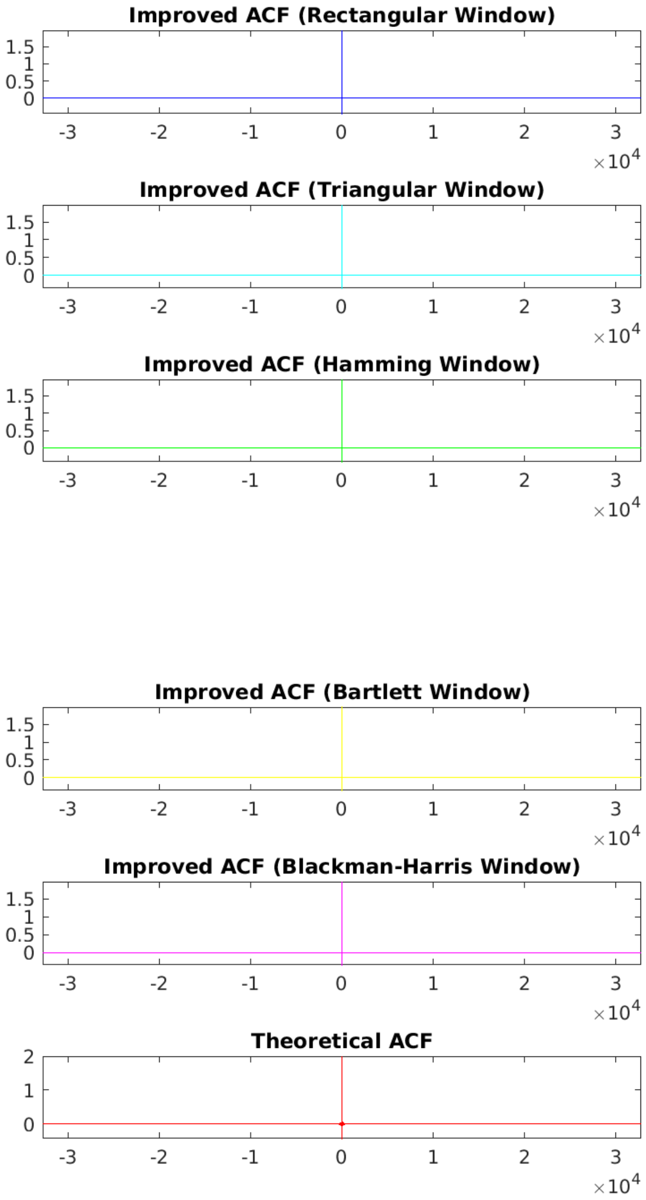

Figure 18: Smoothed ACF

As we can see, the noisy-look we had before has dissappeared, and now the estimated ACF looks more like the theoretical ACF.

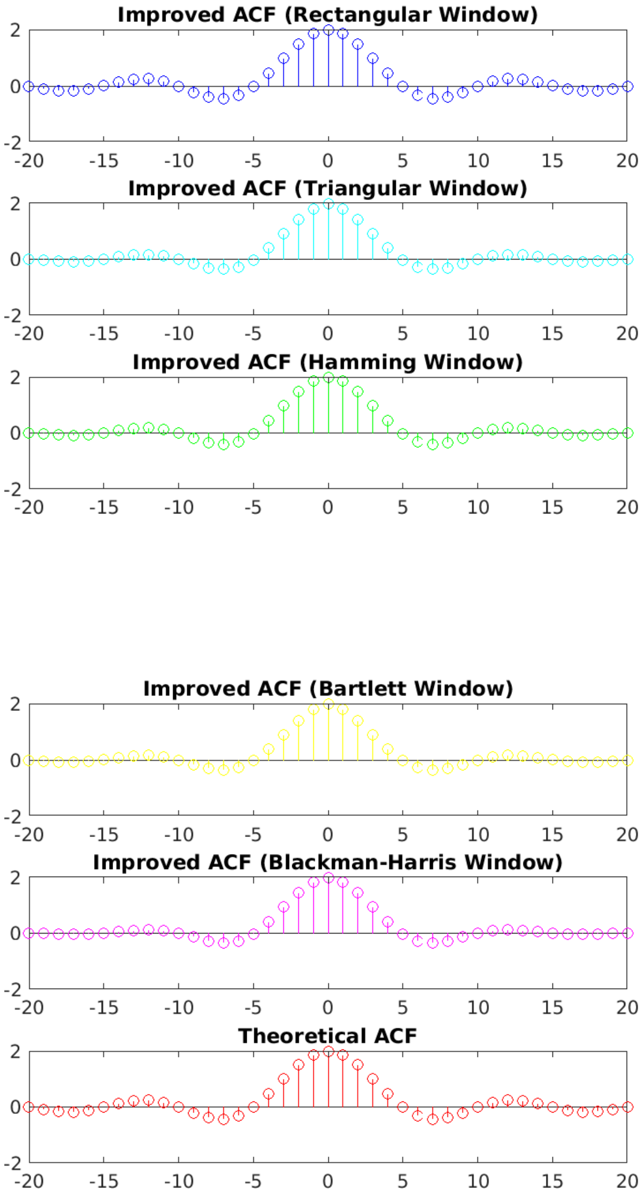

Figure 19: Smoothed ACF

Here there is not much difference, since the estimated graph close up was almost the same as the theoretical graph.

Here we have the PSD:

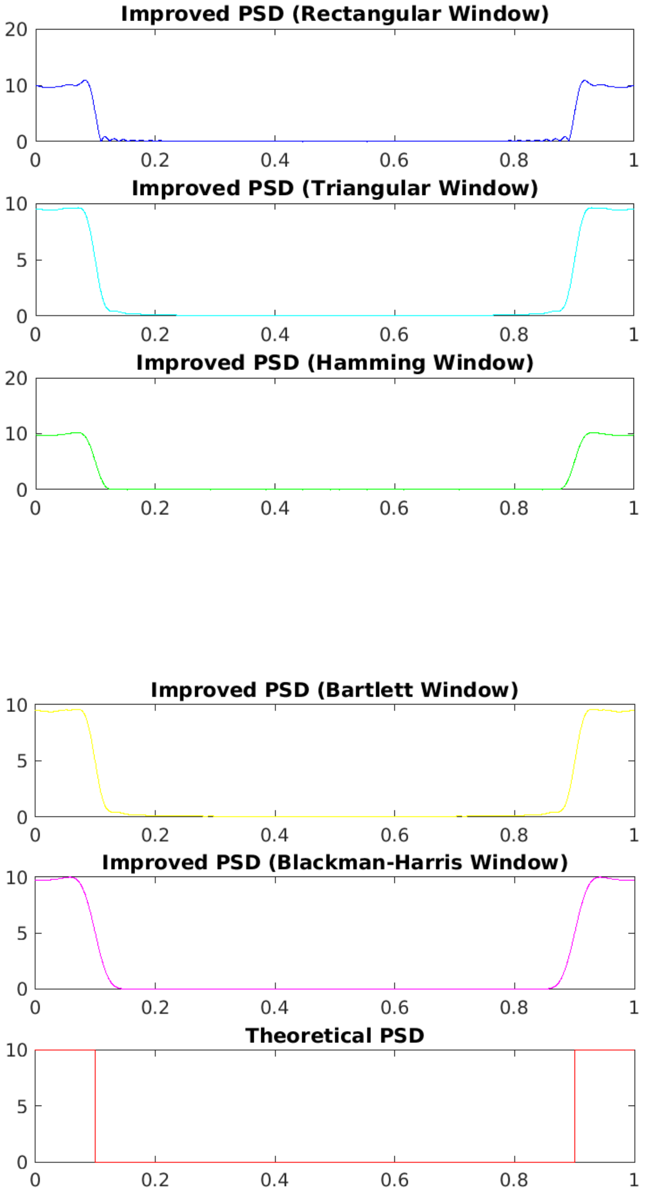

Figure 20: Smoothed PSD

The high variance that we had before is not there anymore, but the edges of the curve still look rounder than they should, and the step is not abrupt enough.

2.2.2 Low-degree Filter:

Here we have the ACF:

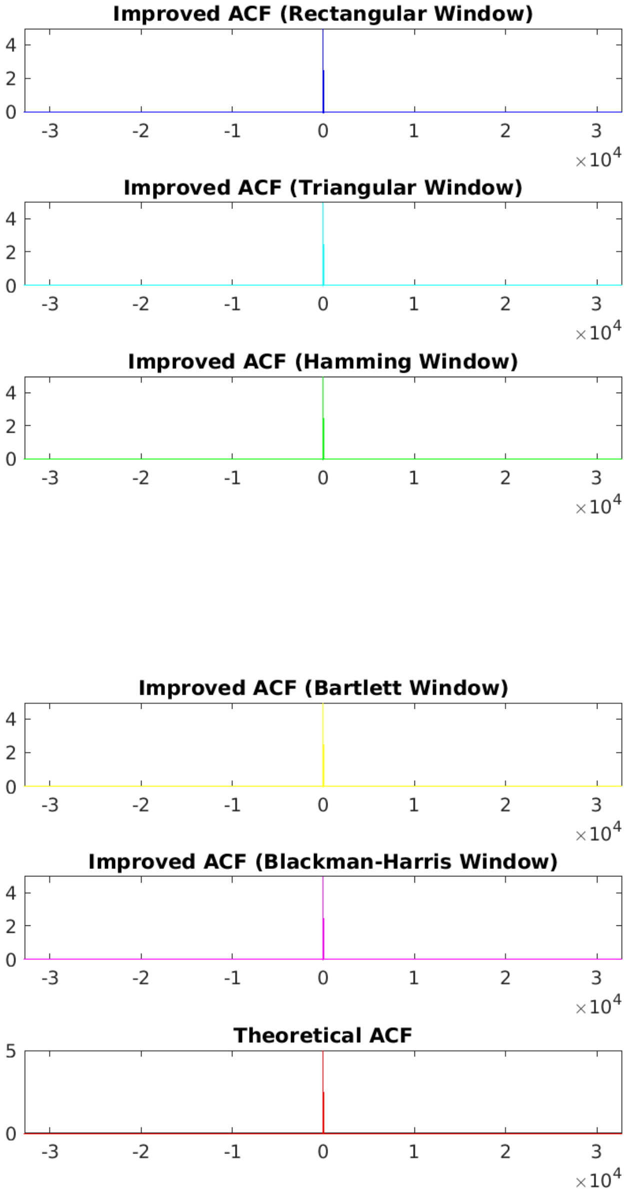

Figure 21: Smoothed ACF

This is the same case as the high-degree filter, now the graph has less variance in respect of the x-axis.

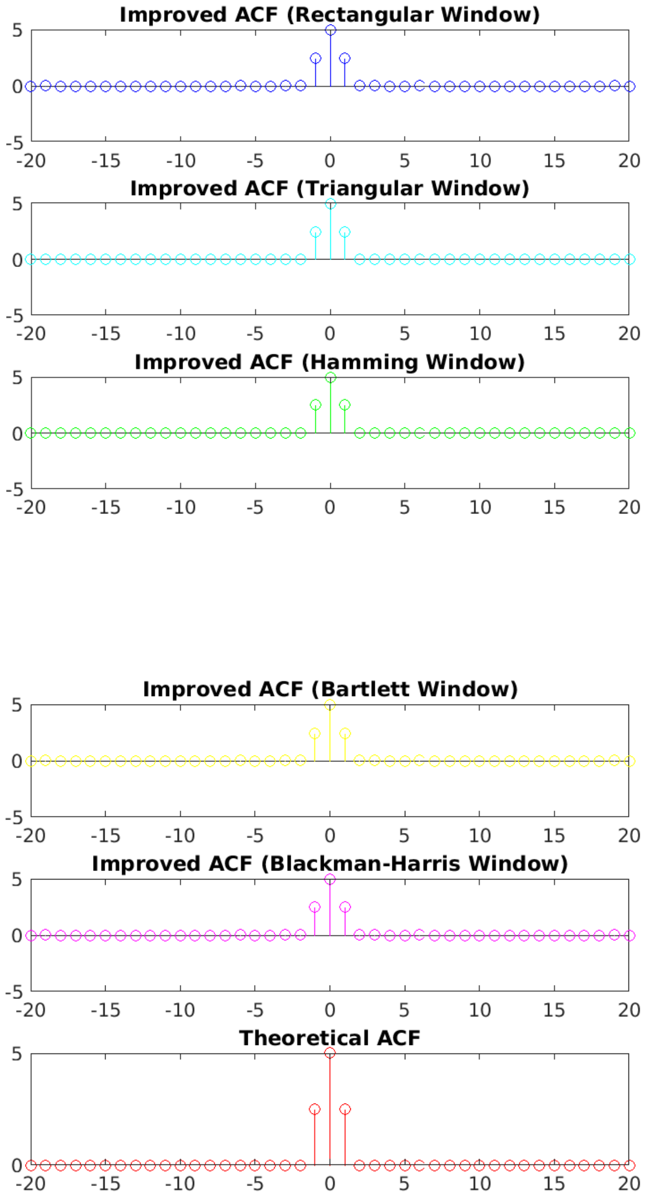

Figure 22: Smoothed ACF

Here, again, there is not much difference, our estimation was very accurate already.

Here we have the PSD:

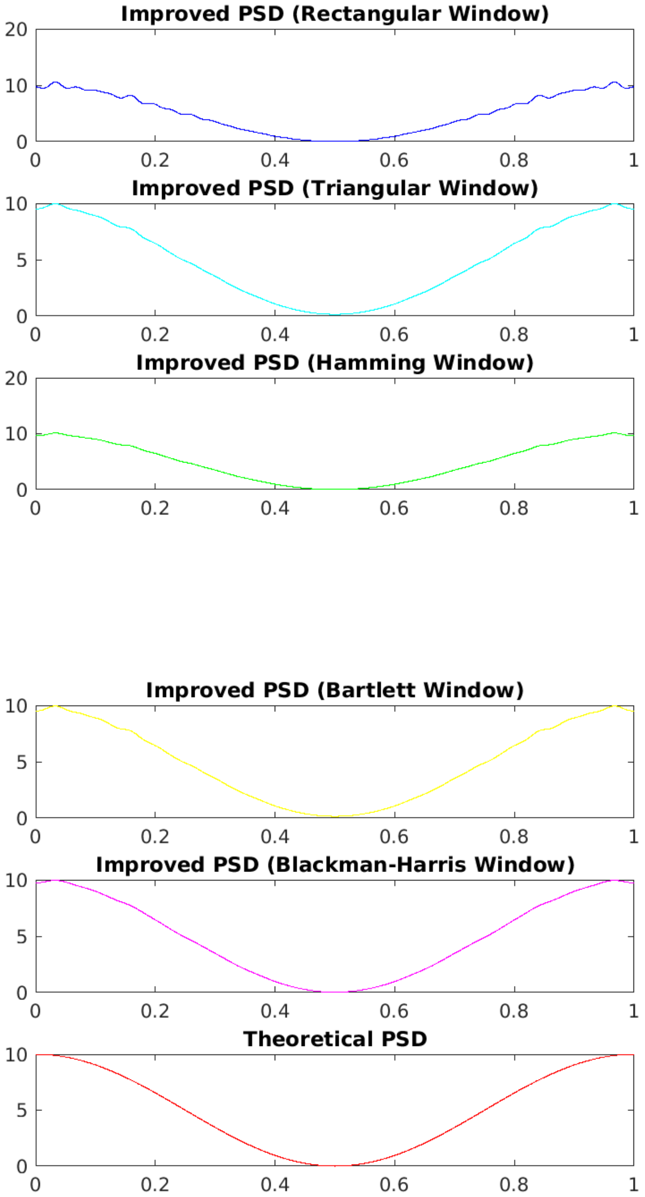

Figure 23: Smoothed PSD

In this PSD we don't have sharp edges, therefore the improvements fit better the theoretical graph than how they did for they high-degree filter. Now we only have a bit of variance, specially with the Rectangular and Triangular window, but with the Blackman-Harris window, the improvement is almost perfect.

2.3 Averaging:

Now let's move on to the improvements through averaging:

2.3.1 High-degree Filter:

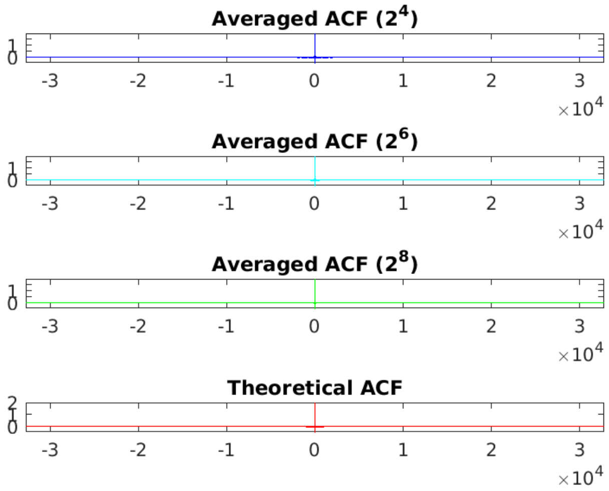

Figure 24: Averaged ACF

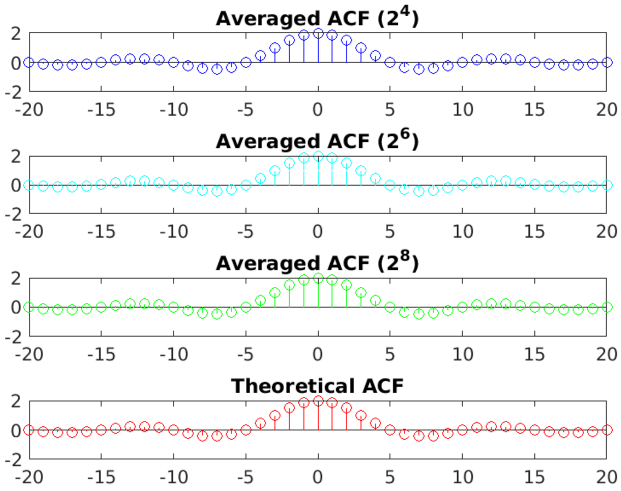

Figure 25: Averaged ACF

We have the same situations as with smoothing: the ACF estimation was very accurate already, so we can not see big changes. Nevertheless, we can notice how in the plotted graph, the noise around 0 decreases as we have higher values for the averaging, since it becomes more accurate.

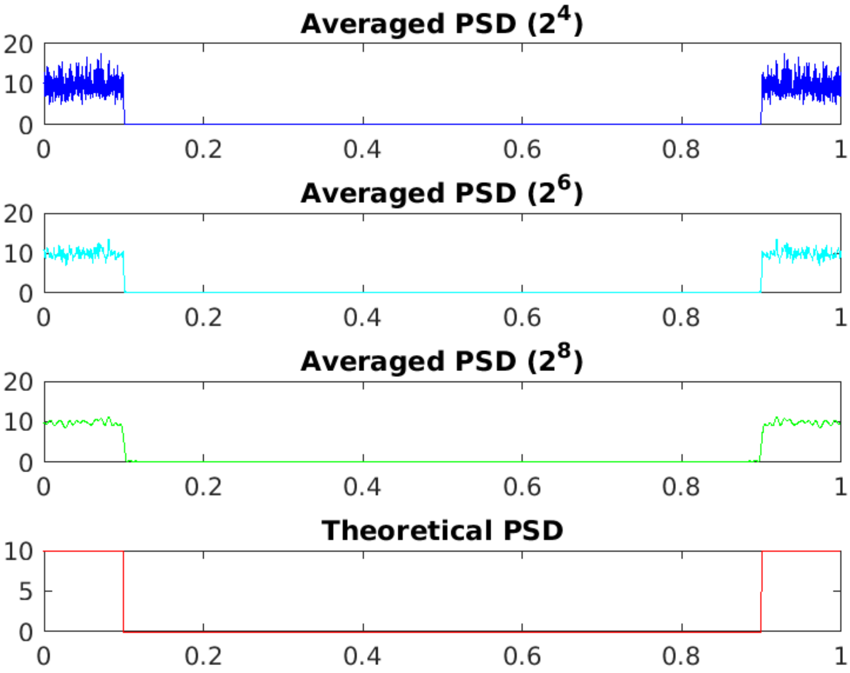

Figure 26: Averaged PSD

Here we can clearly see the changes. The lower the value, the more it still looks like the raw periodogram. Nevertheless, we only go up to 2^8 because if the averaging values get near 2^{16}, that is, our number of samples, the averaged periodogram becomes distorted.

This improvement is better for the PSD since it respects the sharp edges of the rectangle function, and therefore makes the estimation fit better the theoretical graph.

2.3.2 Low-degree Filter:

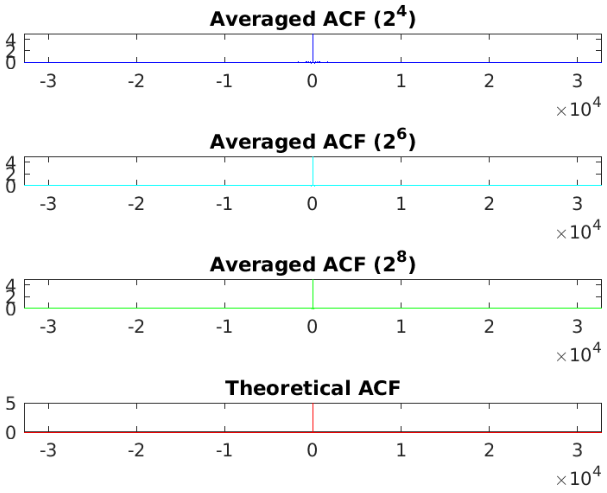

Figure 27: Averaged ACF

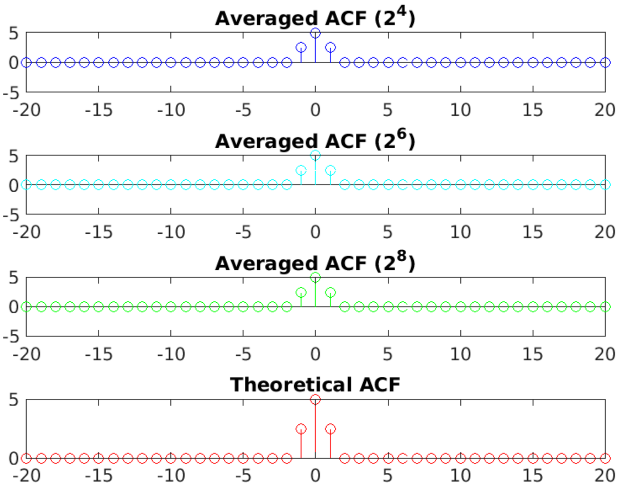

Figure 28: Averaged ACF

Again, we can notice how in the plotted graph, the noise around 0 decreases as we have higher values for the averaging, but it decreases at a higher rate than it did with the high-degree filter.

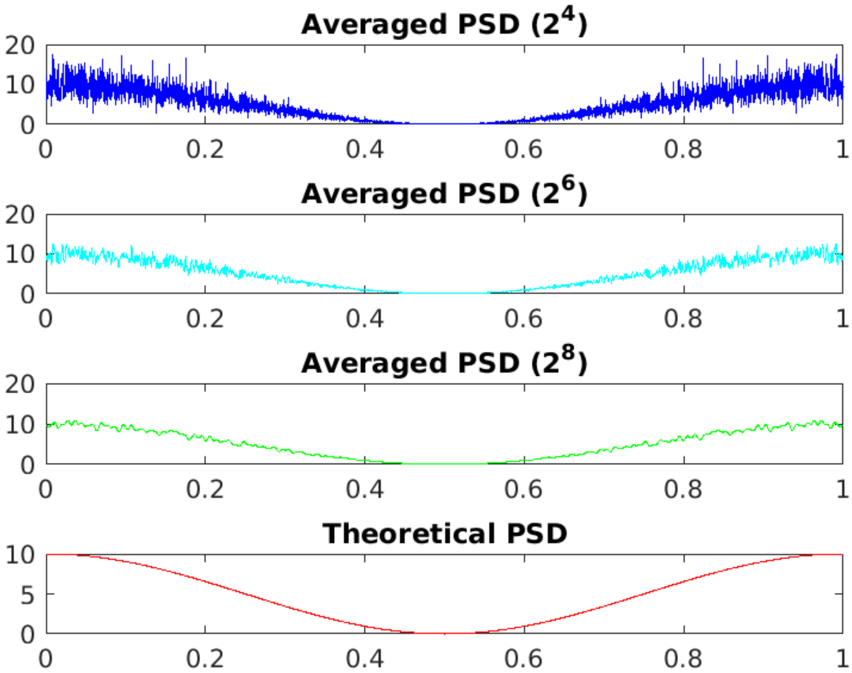

Figure 29: Averaged PSD

The improved graph is good, now we don't have that much variance and the highest value is really similar to the theoretical graph, but the improvements we did with windows were more accurate and less "irregular".

Therefore, if we have a smooth and curvy graph, it is better to use windows because it will be more accurate, but if we have a graph with sharp edges, working with windows might make them appear smoother than we would like to, so it is better to work with averaging because even if it looks a bit noisy still, the shape in the end will be more similar to the theoretical one.

3 Study 3:

3.1 Theoretical Background:

We have the three following systems:

- A squarer:

$$Y[n] = X^2[n] \tag{13}$$

- A half-wave rectifier:

$$Y[n] = \begin{cases} X[n], & n : X[n] > 0 \\ 0, & n : X[n] \leq 0 \end{cases} \tag{14}$$

- An AM-SC modulator:

$$Y[n] = X[n]cos(\omega_{carrier}n) \tag{15}$$

Each one of those is a non-LTI system, that is, even if out input signal is Gaussian and linear, the output doesn't have to be (and probably won't be) Gaussian and linear.

We will use White Gaussian Noise and our high-degree low-pass filter to analyze how these three systems react to out input, and to see how we can spot if a system is LTI or non-LTI.

None of them following a linear equation means that we cannot use our regular formula (3) to calculate the PSD. We have to use special formulas that we can find in the course book. Let's move on to the special calculations for each system.

3.2 Theoretical Analysis:

3.2.1 Squarer:

For the squarer, we can check on the book that the ACF formula is:

$$r_Y[k] = 2r_X^2[k] + r_X^2[0] \tag{16}$$

To calculate the PSD, we just have to swith to frequency domain:

$$\begin{aligned} R_Y[\theta] = \mathcal{F}\{r_Y[k]\} = \\ = 2(R_X * R_X)[\theta] + r_X^2[0]\sum_m \delta(\theta) = \\ = \frac{1}{4}r_X^2[0]\delta(\theta) + 4R_X triangle(\frac{\theta}{2BW}) \end{aligned} \tag{17}$$

The resulting graph is:

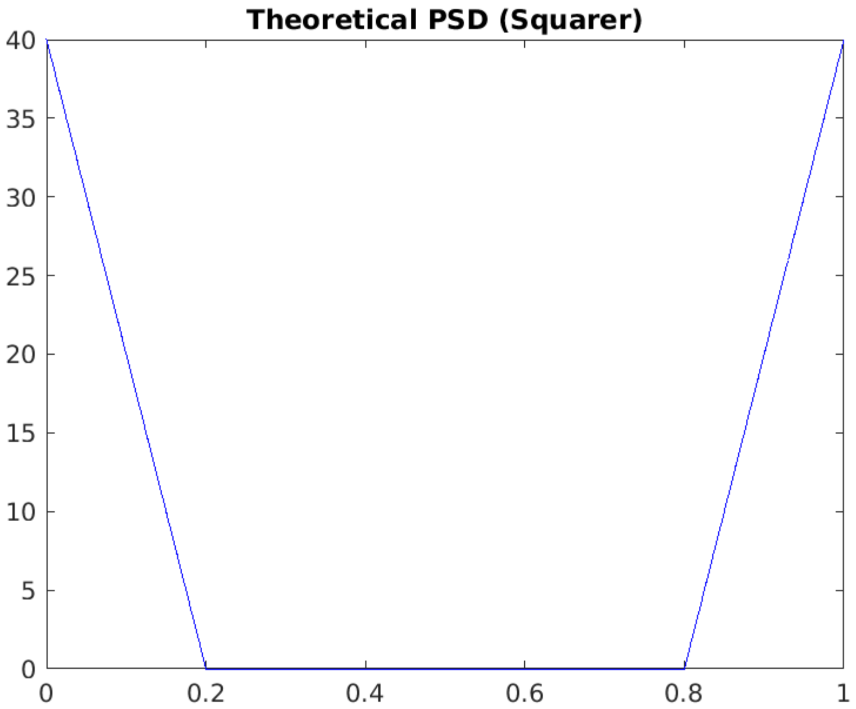

Figure 30: Theoretical PSD

3.2.2 Half-Wave Rectifier:

For the half-wave rectifier, we can check on the book that the ACF formula is:

$$r_Y[k] = \frac{r_X[0]}{2\pi} + \frac{r_X[k]}{4} + \frac{r_X^2[k]}{4\pi r_X[0]} \tag{18}$$

We switch to the frequency domain:

$$\begin{gathered} R_Y[\theta] = \mathcal{F}\{r_Y[k]\} = \\ = R_Y[\theta] = \frac{r_X[0]}{2\pi}\delta[\theta] + \frac{R_X[\theta]}{4} + \frac{(R_X * R_X)[\theta]}{4\pi r_X[0]} = \\ = \frac{R_X}{4\pi}\delta[\theta] + \frac{R_X}{4\pi} triangle[\frac{\theta}{2BW}] + \frac{R_X}{4} rect[\frac{\theta}{2BW}] \end{gathered} \tag{19}$$

The resulting graph from applying that equation is:

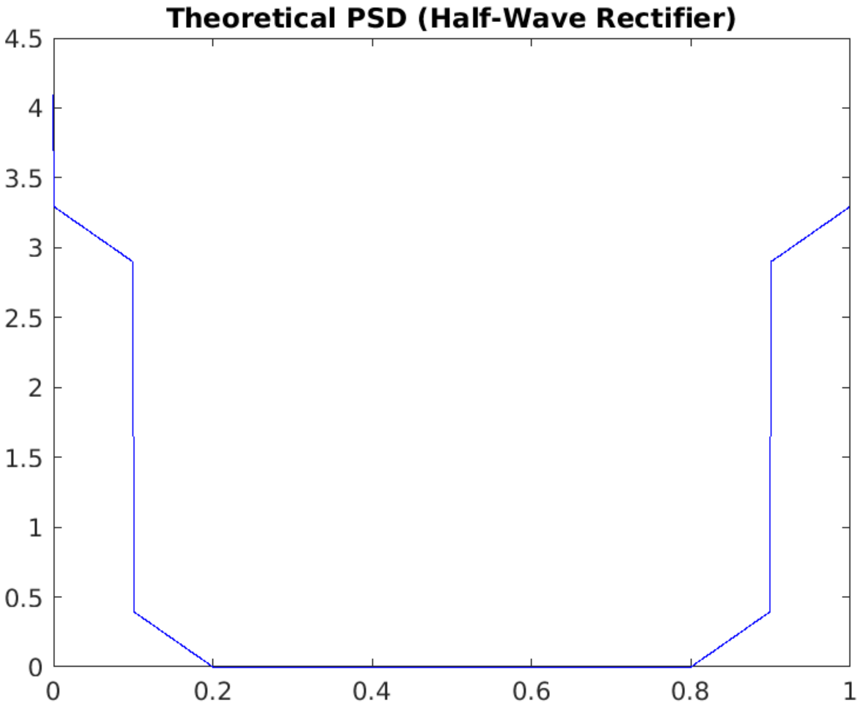

Figure 31: Theoretical PSD

3.2.3 AM-SC Modulator:

For the AM-SC Modulator, we are multiplying our signal by a carrier. Our carrier will have aplitude $A = 1$ and frequency $f_{carrier} = 0.25$.

$$Carrier[k] = cos(\omega_{carrier} k) \tag{20}$$

We can check on the book that the ACF formula is:

$$r_Y[k] = \frac{A^2}{2} cos(2\pi f_{carrier} k) = \frac{A^2}{2} cos(\omega_{carrier} k) \tag{21}$$

We switch to the frequency domain:

$$\begin{gathered} R_Y[\theta] = \mathcal{F}\{r_Y[k]\} = \\ = \frac{R_X}{4}(rect[\frac{\theta - f_{carrier}}{BW}] + rect[\frac{\theta + f_{carrier}}{BW}]) \end{gathered} \tag{22}$$

The resulting graph from applying that equation is:

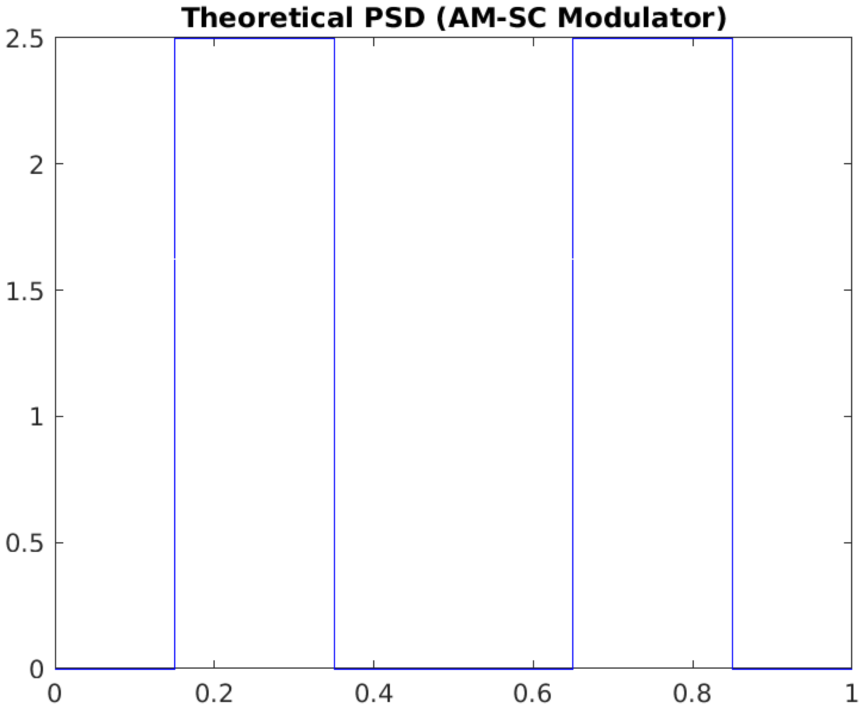

Figure 32: Theoretical PSD

3.3 Estimations:

3.3.1 Squarer:

For the squarer, we get this PSD estimation:

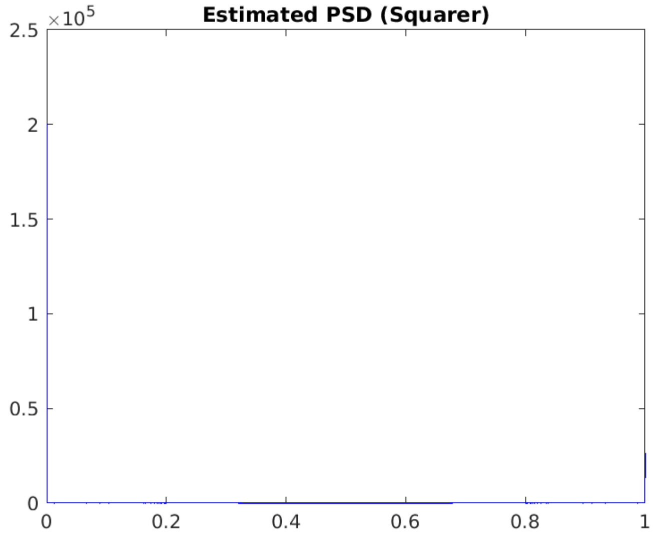

Figure 33: Estimated PSD

Due to the non-linearity of the system, it has a peak in $f = 0$, so we can't really see what's going on in the lower values of the y-axis. If we zoom, we see something that looks more like the PSD we are looking for:

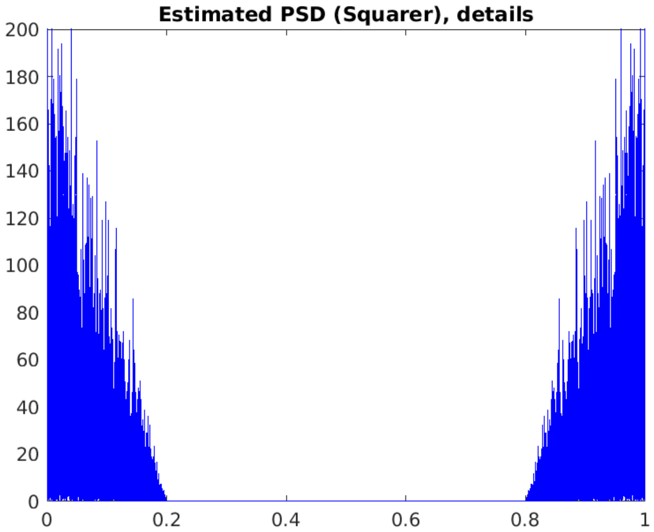

Figure 34: Zoomed Estimated PSD

We can compare it to the estimated PSD to see that, as every PSD we have seen until now, it has a lot of variance, but we can see that it corresponds with the function we already analyzed:

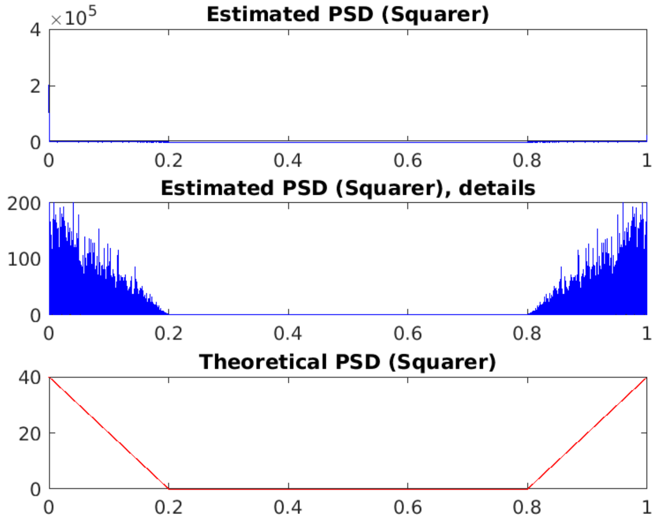

Figure 35: Compared PSD

Lastly we can see the histogram of the squared signal:

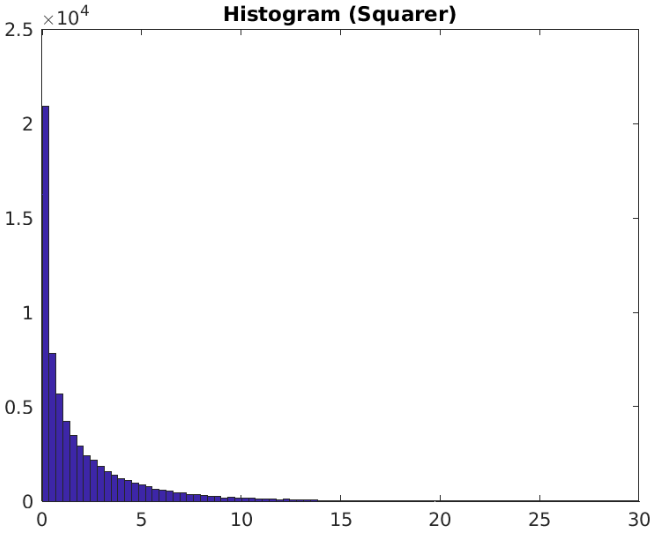

Figure 36: Histogram

We will analyze this in comparison to the initial filtered signal histogram further on.

3.3.2 Half-Wave Rectifier:

For the half-wave rectifier, we get this PSD estimation:

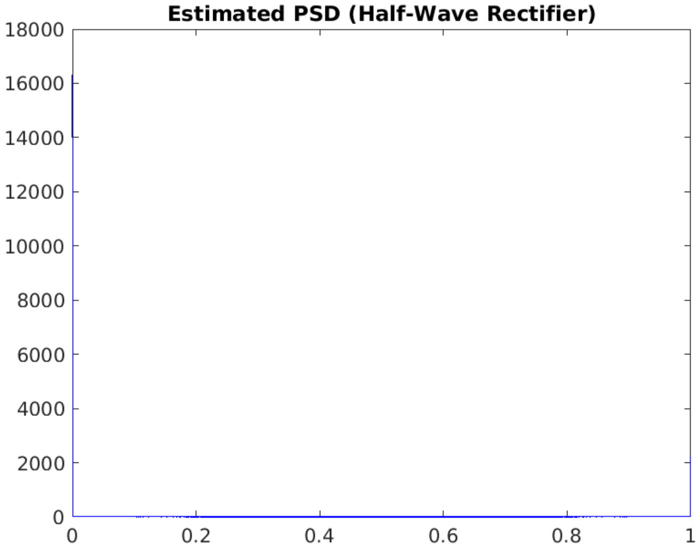

Figure 37: Estimated PSD

Again, as the squarer PSD, it has a peak in $f = 0$. Therefore, we will zoom the y-axis in the lower values to check:

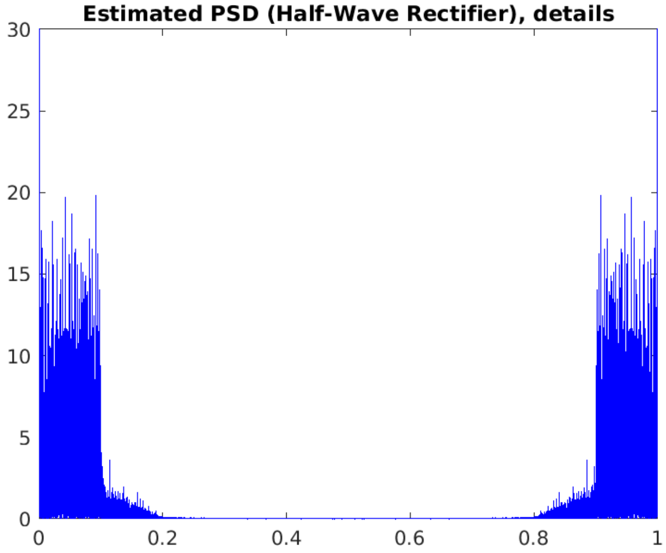

Figure 38: Zoomed Estimated PSD

We can compare it to the estimated PSD to see that, as every PSD we have seen until now, it has a lot of variance, but we can see that it corresponds with the function we already analyzed:

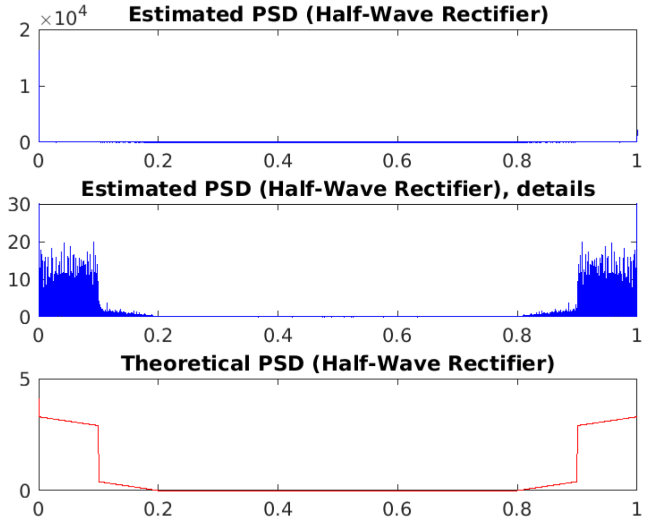

Figure 39: Compared PSD

Lastly we can see the histogram of the rectified signal:

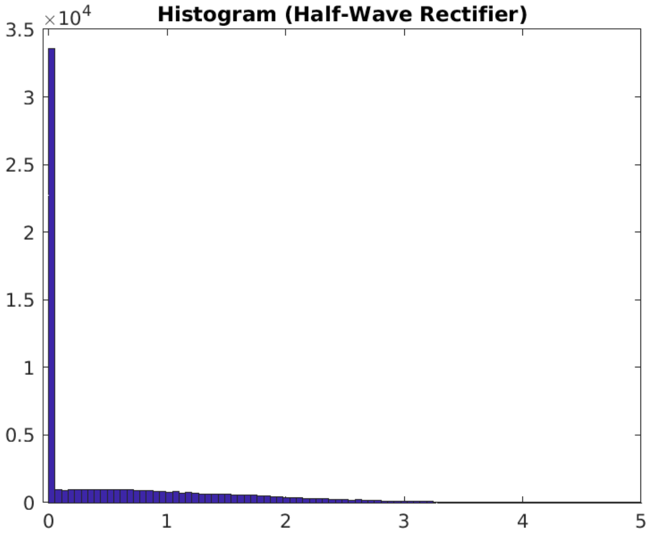

Figure 40: Histogram

3.3.3 AM-SC Modulator:

For the AM-SC Modulator, we get this PSD estimation:

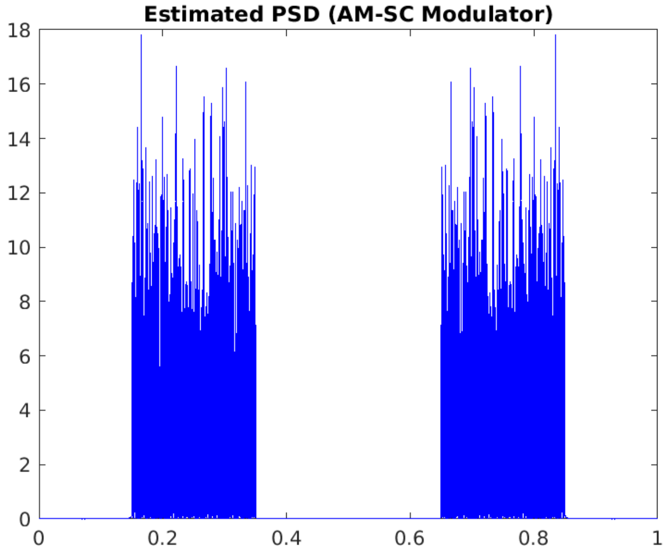

Figure 41: Estimated PSD

We can compare it to the estimated PSD to see that, as every PSD we have seen until now, it has a lot of variance, but we can see that it corresponds with the function we already analyzed:

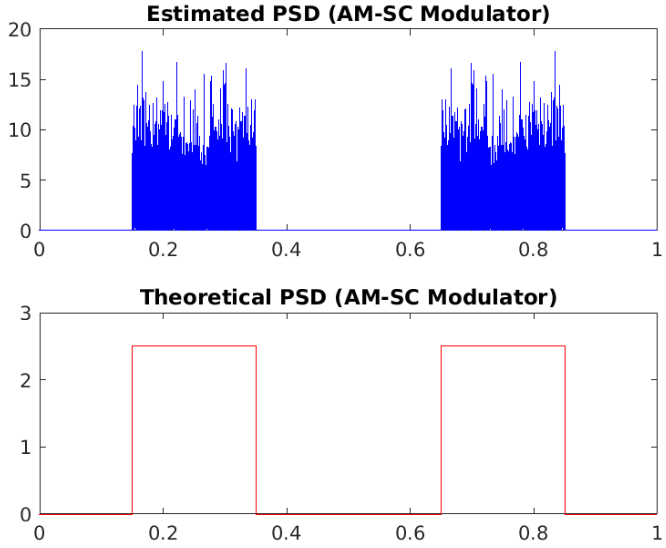

Figure 42: Compared PSD

Lastly we can see the histogram of the squared signal:

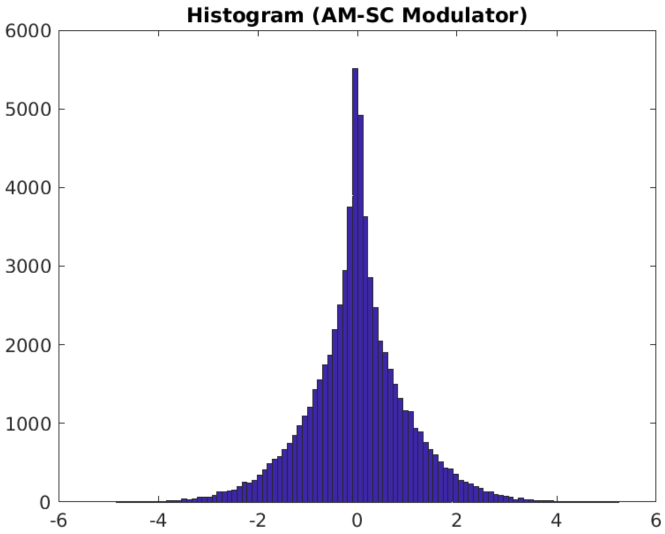

Figure 43: Histogram

3.4 Comparisons:

Here we can see how the theoretical and estimated PSDs change depending on the system we used and regarding the original filtered signal:

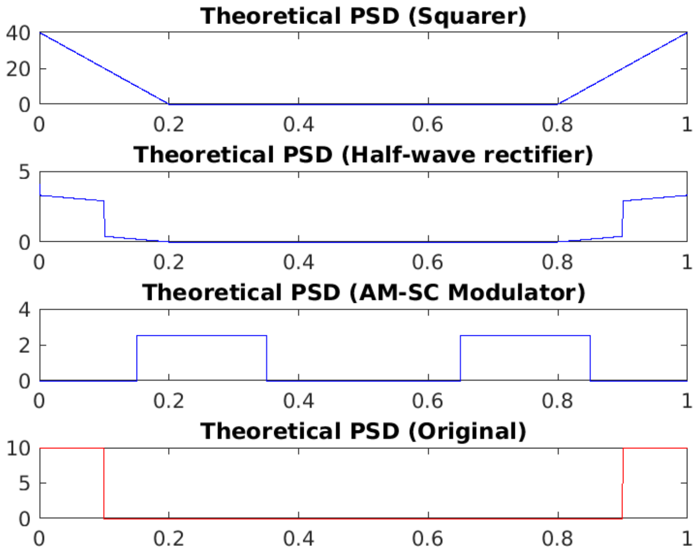

Figure 44: Compared Theoretical PSDs

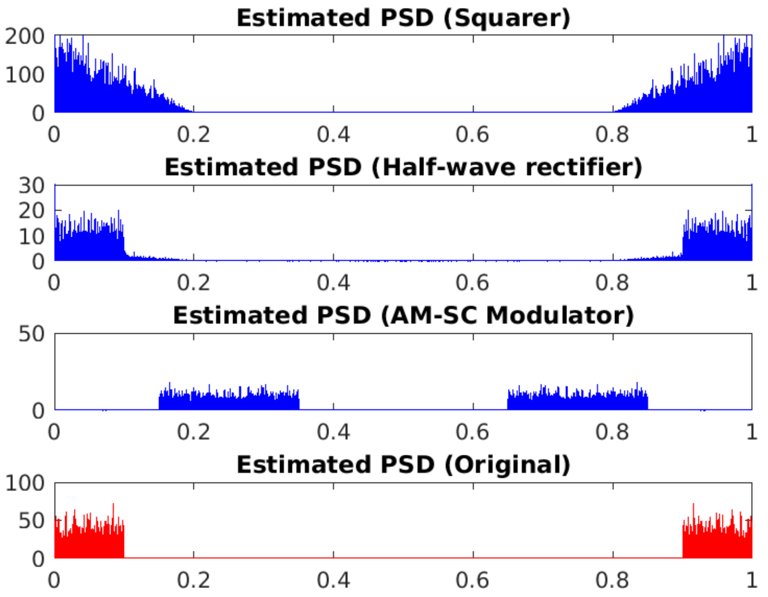

Figure 45: Compared Estimated PSDs

Here we can analyze the different histograms:

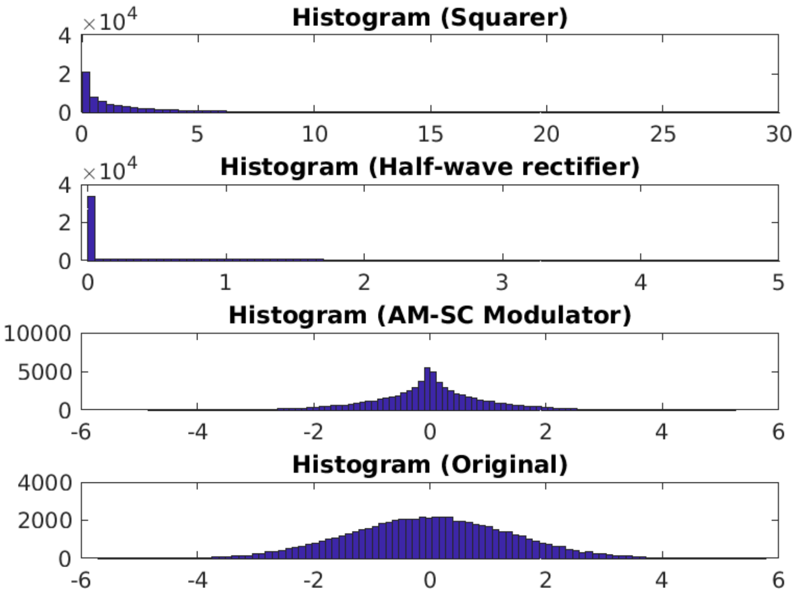

Figure 46: Compared Histograms

We can see that the systems we studied present non-linearities in the estimated PSDs, because we get that peak near $f = 0$ that was not there in the original signal.

Furthermore, we can also see the non-linearities looking at the histograms: our signal was Gaussian, and that is easy to see on the histogram of the figure. We know that a Gaussian input that goes through a LTI system always gives a Gaussian output. Nevertheless, if we look at the other histograms, we can appreciate that the Gaussian kind of graph has dissappeared. Therefore, if we didn't know that the three systems we applied are non-linear, we would have been able to get that information from the histograms.

4 Study 4:

4.1 Theoretical Background:

For this study we will be manipulating the signal in ways that are commonly used in telecommunication areas.
We have two special operations:

$$Y[n] = X[n](-1)^n \tag{23}$$

$$Y[n] = \begin{cases} X[n], & n : odd \\ 0, & n : even \end{cases} \tag{24}$$

Again, we will use our White Gaussian Noise and a high-degree low-pass filter approximated as an ideal filter.

4.2 Theoretical Analysis:

4.2.1 First System:

Refering to formula (23), we will have a system that takes our signal and multiplies every value by either $+1$ or -1, alternating that sequence.
We have to calculate the PSD from the ACF:

$$\begin{gathered} r_Y[k] = E\{Y[n+k]Y[n]\} = E\{X[n+k]H[n+k]X[n]H[n]\} = \\ E\{X[n+k](-1)^{n+k}X[n](-1)^n\} = E\{X[n+k]X[n](-1)^{2n+k}\} = \\ E\{X[n+k]X[n](-1)^k\} = E\{X[n+k]X[n]\}(-1)^k = r_X[k](-1)^k \end{gathered} \tag{25}$$

Now to go to the PSD we have to do the Fourier Transform of the above:

$$\begin{gathered} R_Y[\theta] = \mathcal{F}\{r_Y[k]\} = \\ = \mathcal{F}\{r_X[k](-1)^k\} = R_X[\theta - 0.5] = \\ = R_X rect(\frac{\theta - 0.5}{BW}) \end{gathered} \tag{26}$$

This is the result:

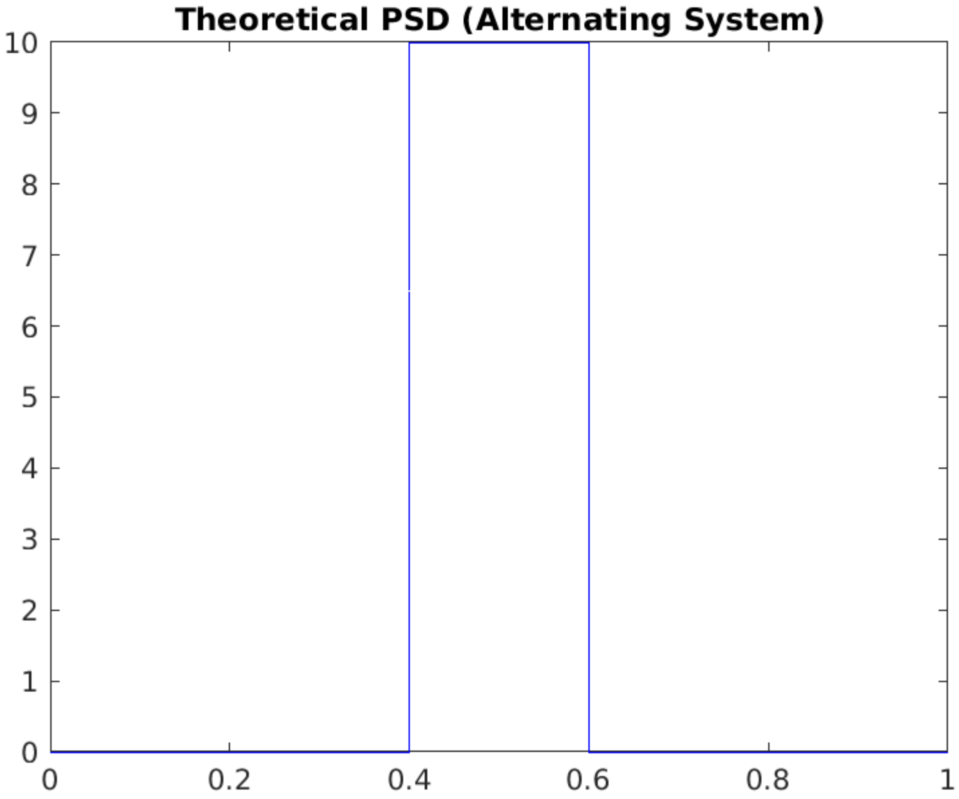

Figure 47: Theoretical PSD

4.2.2 Second System:

Refering to formula (24), we will have a system that takes every other value of our signal and multiplies it by 0, so we will have a sequence where one every two values is 0 and the other is the actual value from out original filtered signal. If we translate from formula (24) to a single-line formula, we get:

$$Y[n] = \frac{1}{2}X[n](1 - (-1)^{n+A}) \tag{27}$$

A is a random variable with $A \in [0, 1]$ with the same probability of being 0 or 1. That way, $(-1)^{n+A}$ can be either 0 or 2, and in case that it is 2, we have $\frac{1}{2}$, so in the end we decimate the signal.

Again, we have to calculate the PSD from the ACF:

$$\begin{aligned}
r_Y[k] = E\{Y[n+k]Y[n]\} = E\{\frac{1}{2}X[n+k](1-(-1)^{n+k+A})\frac{1}{2}X[n](1-(-1)^{n+A})\} = \\
= E\{X[n+k]X[n]\frac{1}{4}(1-(-1)^{n+k+A})(1-(-1)^{n+A})\} = \\
= \frac{1}{4}E\{X[n+k]X[n](1-(-1)^{n+k+A}-(-1)^{n+A})+(-1)^{n+k+A+n+A})\} = \\
= \frac{1}{4}E\{X[n+k]X[n](1-(-1)^{n+k+A}-(-1)^{n+A})+(-1)^{k})\} = \\
= \frac{1}{4}E\{X[n+k]X[n](1-(-1)^{n+k+A}-(-1)^{n+A})+(-1)^{k})\} = \\
= \frac{1}{4}[E\{X[n+k]X[n]\}-E\{X[n+k]X[n](-1)^{n+k+A}\}-E\{X[n+k]X[n](-1)^{n+A}\}+E\{X[n+k]X[n](-1)^{k}\}] = \\
= \frac{1}{4}[E\{X[n+k]X[n]\}+E\{X[n+k]X[n]\}(-1)^{k}] = \frac{1}{4}[r_X[k]+r_X[k](-1)^{k}]
\end{aligned} \tag{28}$$

Now to go to the PSD we have to do the Fourier Transform of the above:

$$\begin{aligned}
R_Y[\theta] = \mathcal{F}\{r_Y[k]\} = \mathcal{F}\{\frac{1}{4}[r_X[k]+r_X[k](-1)^{k}]\} = \\
= \frac{1}{4}(R_X[\theta-0.5]+R_X[\theta]) = \frac{R_X}{4}(rect(\frac{\theta-0.5}{BW})+rect(\frac{\theta}{BW})
\end{aligned} \tag{29}$$

This is the result:

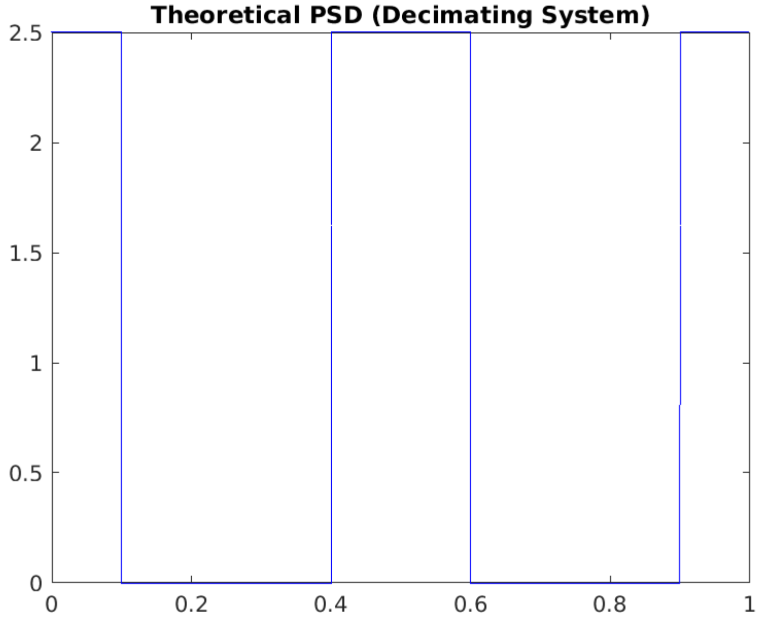

Figure 48: Theoretical PSD

4.3 Estimations:

Now we will move to the estimatied PSDs:

4.3.1 First System:

For the first system we get this estimated PSD:

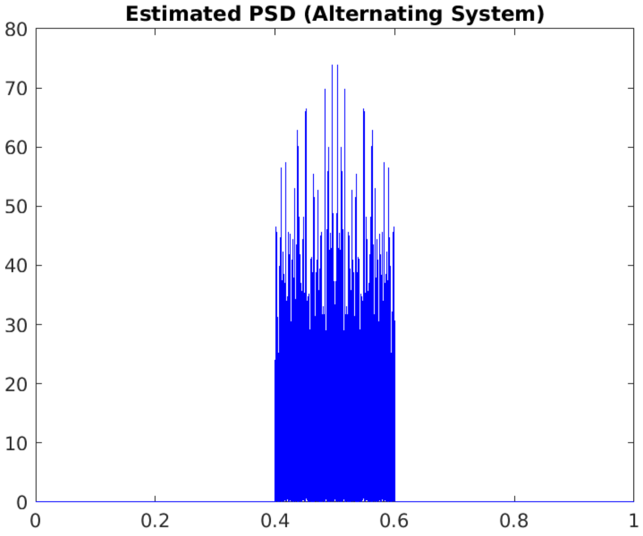

Figure 49: Estimated PSD

If we compare it to the theoretical PSD, we can see the similarities:

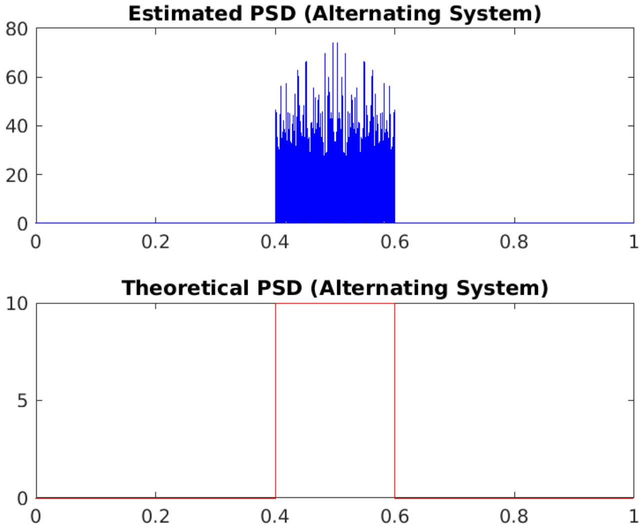

Figure 50: Compared PSDs

4.3.2 Second System:

For the second system, the resulting graph is:

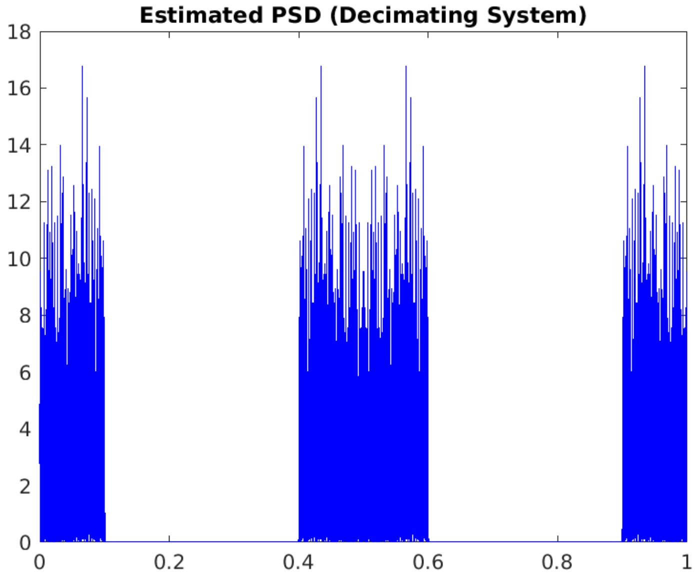

Figure 51: Estimated PSD

We can compare it to the theoretical PSD too:

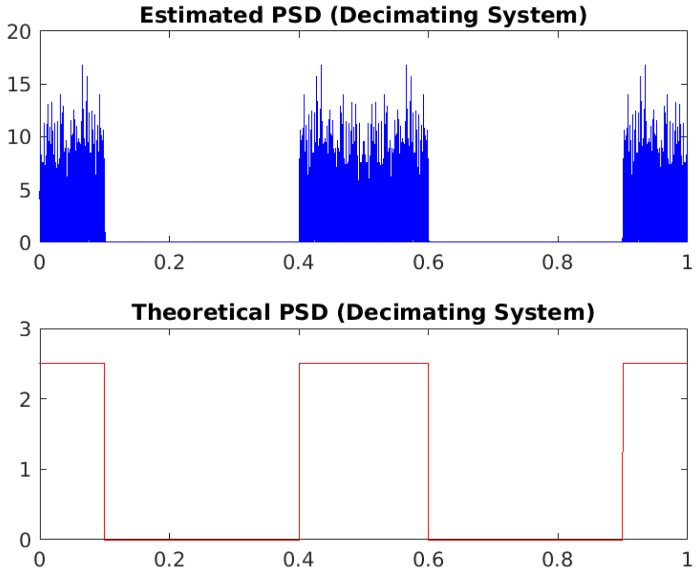

Figure 52: Compared PSDs

4.4 Conclusions:

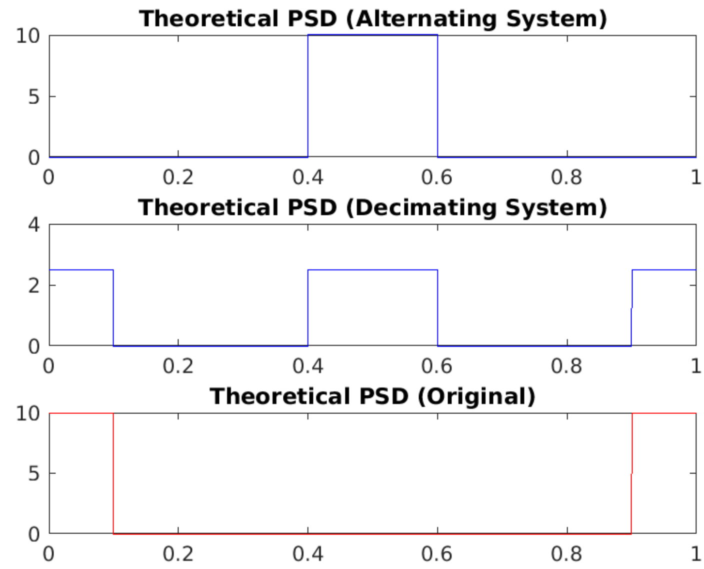

Figure 53: Compared Theoretical PSDs

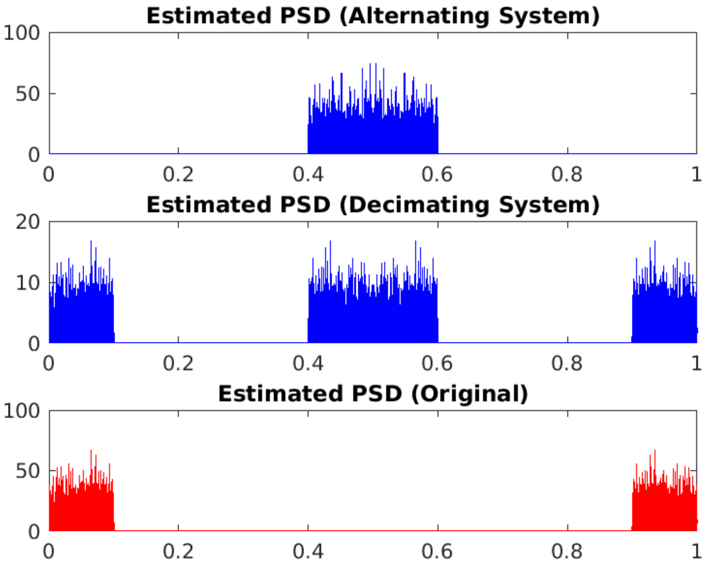

Figure 54: Compared Estimated PSDs

As we can see, in both systems we have the PSD in frequencies that we didn't have in the input filtered signal. That mean that the systems are not LTI, like the systems in study 3.