3TR4: Communication Systems
Lab 4

Instructor: Dr. Chen

Tamer Rafidi – L09 – 400333527
Samer Rafidi – L09 – 400333524

Numerical Experiment 1

i) Calculate Theoretical Function

The PSD equation is derived from the Fourier transform of the autocorrelation function.

$$S_y(f) \ = \ k\ |H(f)|^2$$

$$= \ k * rect(\frac{f}{500})$$

Since we know that $R_y(\tau) \leftrightarrow S_y(f)$, we can find the IFT to solve for Ry

$$rect(t) \leftrightarrow \sin c\ (f)$$

$$\sin c\ (t) \leftrightarrow rect(f)$$

We can use the scaling property

$$\sin c\ (500t) \leftrightarrow \frac{1}{500} rect\left(\frac{f}{500}\right)$$

We then can multiply both sides by 500k

$$500k \sin c\ (500t) \leftrightarrow k \cdot rect\left(\frac{f}{500}\right)$$

Now we can compare this equation to the PSD equation found above to conclude that:

$$R_y(\tau) \leftrightarrow 500k \cdot \sin c\ (500t)$$

Looking at this equation, we can see that we will get zeros at $\pm\frac{1}{500}t$ which means we will get zeros at $\pm 0.002,\ 0.004,\ etc.$

ii) Change the maxlag from 100 to 200 to 500 and comment on PSD

As we increase the maxlag, we see there are more frequency samples between the two peaks. This is because the frequency resolution of the PSD increases as we increase the maxlag.

iii) Estimate bandwidth of the filter using autocorrelation plot

When zooming into the plots from MATLAB, we can see that the zeros are split apart by every 0.002 seconds. This makes sense as that is the value that was expected from our theoretical calculations above. We can determine that the bandwidth is 250Hz as there are peaks and troughs every 0.004 seconds.

Plots used to answer experiment 1

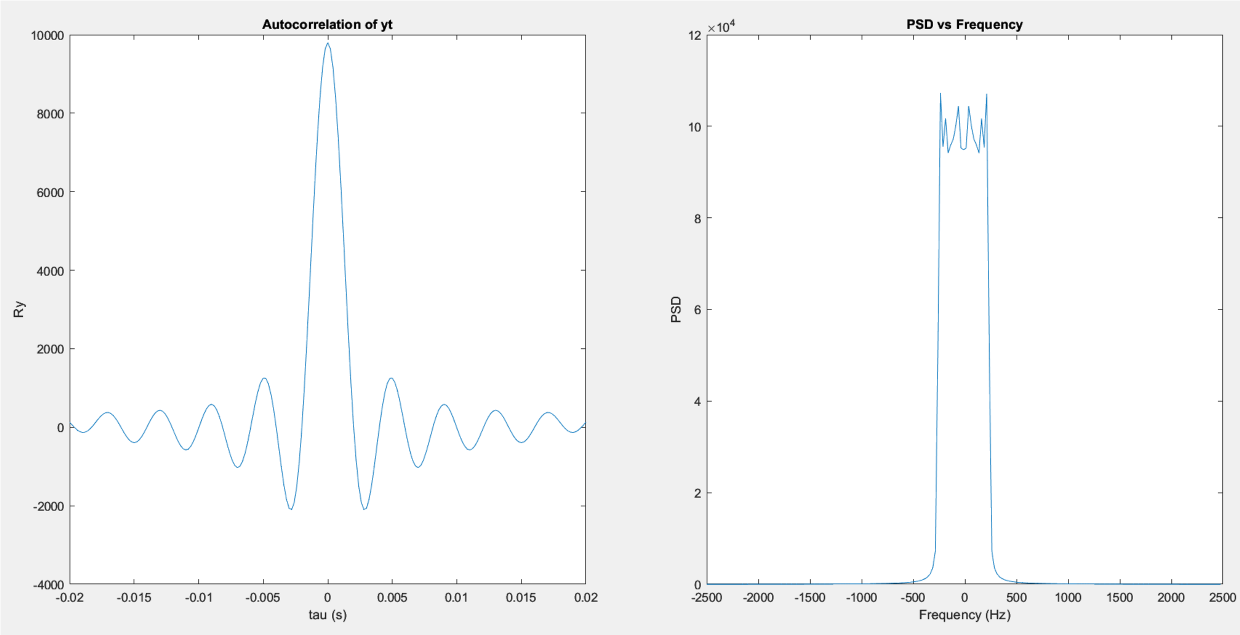


Figure 1. Autocorrelation and PSD at maxlag = 100

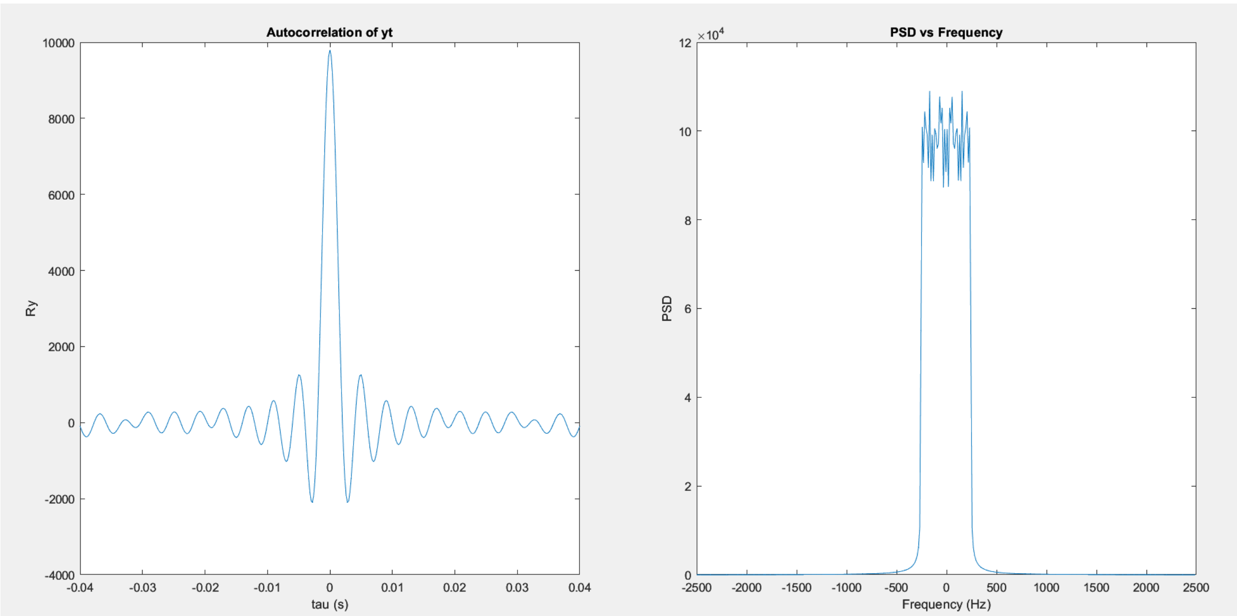


Figure 2. Autocorrelation and PSD at maxlag = 200

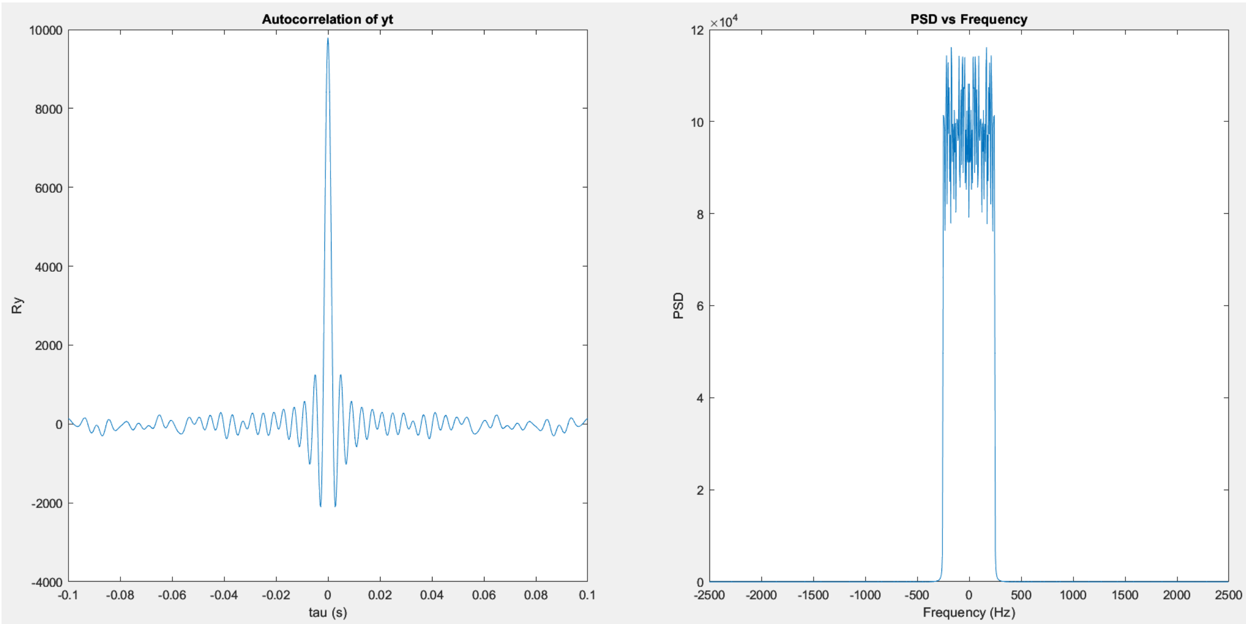


Figure 3. Autocorrelation and PSD at maxlag = 500

Numerical Experiment 2

Theoretical Functions

$$y(t) = X(t) + w(t)$$

$$y(t) = A \cdot \sin(2\pi f_c t + \theta) \; + \; w(t)$$

$$R_y(t) = E\big[\big(A \cdot \sin(2\pi f_c t + \theta) + w(t)\big) \cdot \big(A \cdot \sin(2\pi f_c(t + \tau) + \theta) + w(t + \tau)\big)\big]$$

To avoid writing the variables many times, I will assign an x and y variable to shorten my equation.

$$x = 2\pi f_c t + \theta \; and \; y \; = 2\pi f_c(t + \tau) + \theta$$

$$= E[A^2 \sin(x) \sin(y) + A \sin(x) \cdot w(t + \tau) + w(t) \cdot A \sin(y) + w(t) \cdot w(t + \tau)]$$

$$= E[A^2 \sin(x) \sin(y) + 0 + 0 + w(t) \cdot w(t + \tau)]$$

We can use the trig identity $\sin(x) \sin(y) = \frac{1}{2}(\cos(x - y) - \cos(x + y))$

$$= \frac{A^2}{2} \cdot E[\cos(2\pi f_c t) - \cos(4\pi f_c t + 2\pi f_c(t + \tau) + 2\theta) + k\delta(\tau)]$$

The 2 θ term makes that whole section = 0. So, the final equation is:

$$= \frac{A^2}{2} \cdot \cos(2\pi f_c \tau) + k\delta(\tau)$$

The PSD is derived below.

$$R_y(\tau) = \frac{A^2}{2} \cdot \cos(2\pi f_c \tau) + k\delta(\tau)$$

$$S_y(f) = \frac{A^2}{4} \cdot \big(\delta(f - f_c) + \delta(f + f_c)\big) + k$$

Comparing our theoretical calculations with what out MATLAB produced, we can see that there is an impulse at τ = 0 which is shown in our MATLAB plot. Also, both PSD's have an impulse at -fc and +fc which can be seen in the MATLAB plot.

i) Do I observe a peak at the zero lag in the autocorrelation plot?

Yes, we can see a peak at the zero lag in the autocorrelation plot. This happens because of the impulse in the autocorrelation function caused by the white noise.

ii) Explain the connection between the maxlag and frequency resolution

Looking at the plots, we can see that increasing the maxlag increases the frequency resolution of the PSD. The result of this is a larger peak in the MATLAB PSD plots. We can also see in the table below that the fc values increase as the maxlag increase. We can also see there are more frequency samples between the two peaks (+-Fc).

Maxlag	f_c
100	236.32 Hz
200	243.14 Hz
20000	249.93 Hz

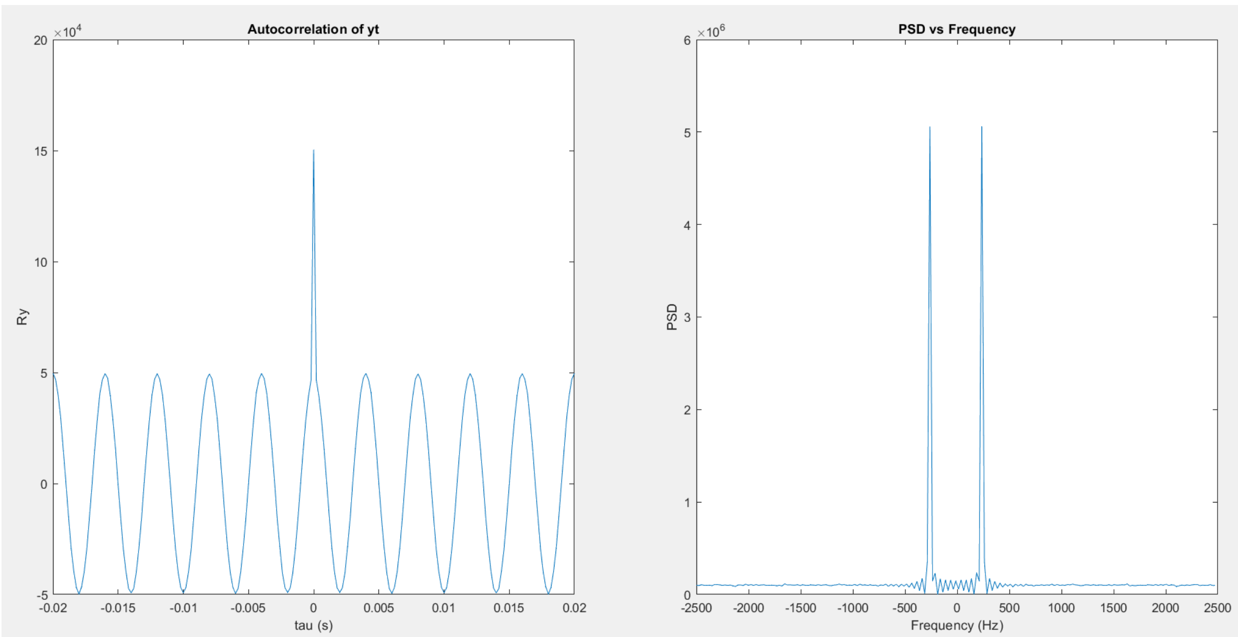


Figure 4. Autocorrelation and PSD at maxlag = 100

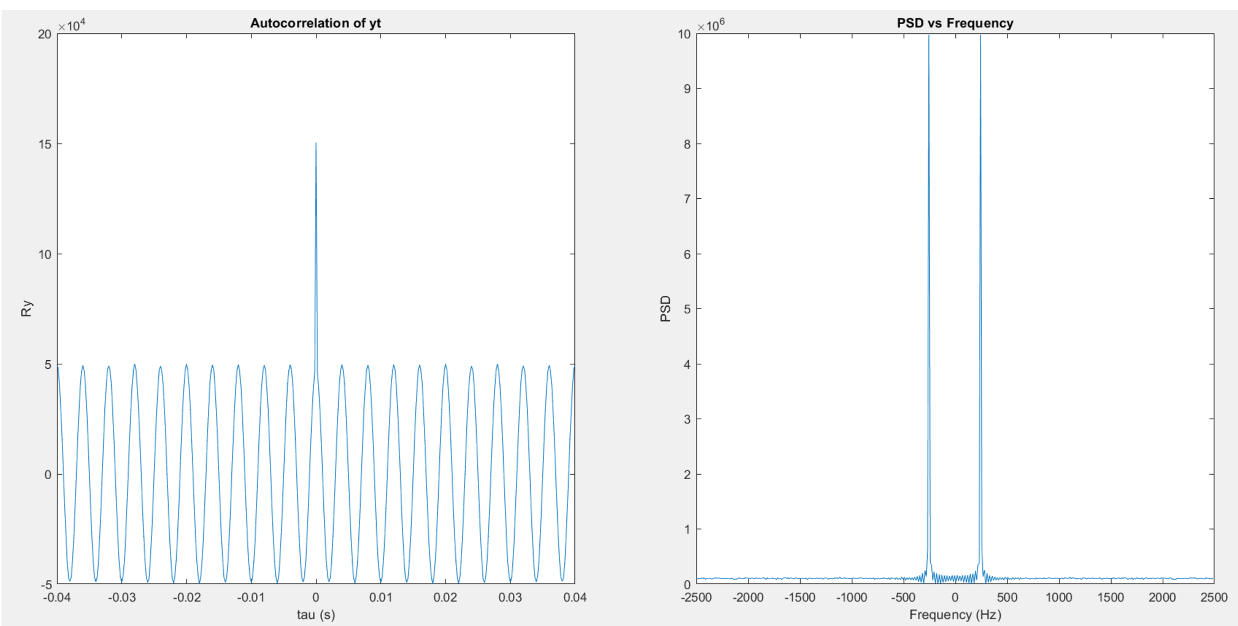


Figure 5. Autocorrelation and PSD at maxlag = 200

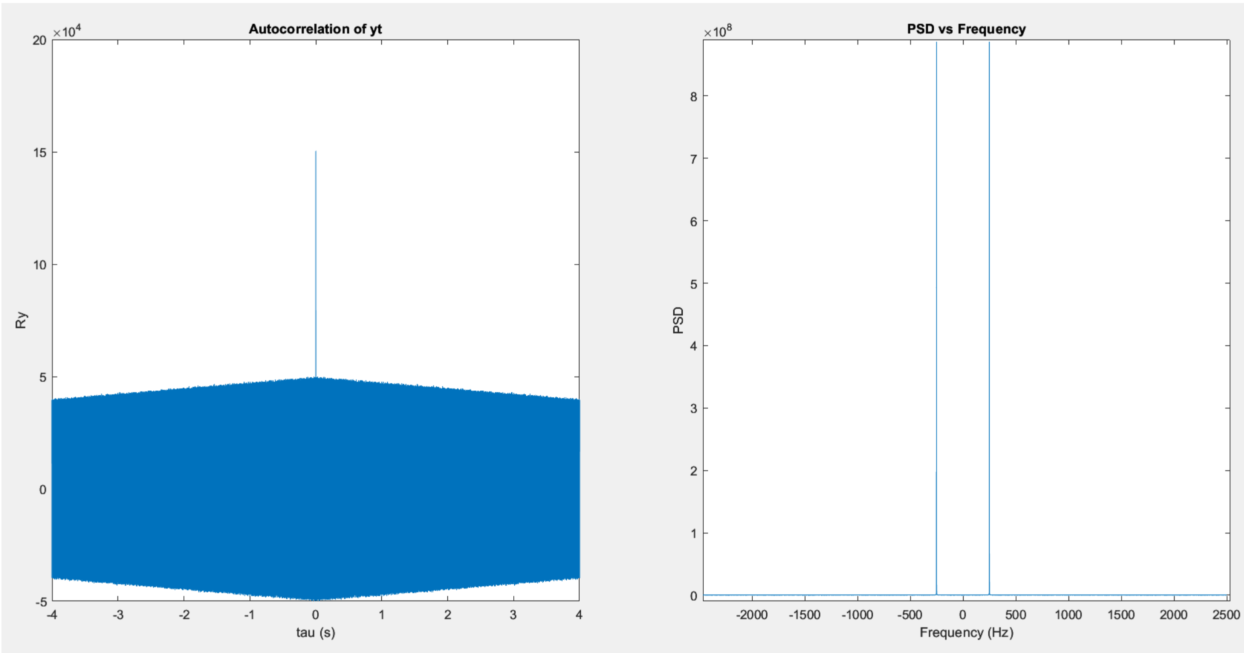


Figure 6. Autocorrelation and PSD at maxlag = 20000

iii) Plotting yt as a function of time

No, I cannot estimate the frequency directly from the plot. There is too much noise for me to understand what I am looking at. I am expecting a sinusoidal graph but there is too much noise.

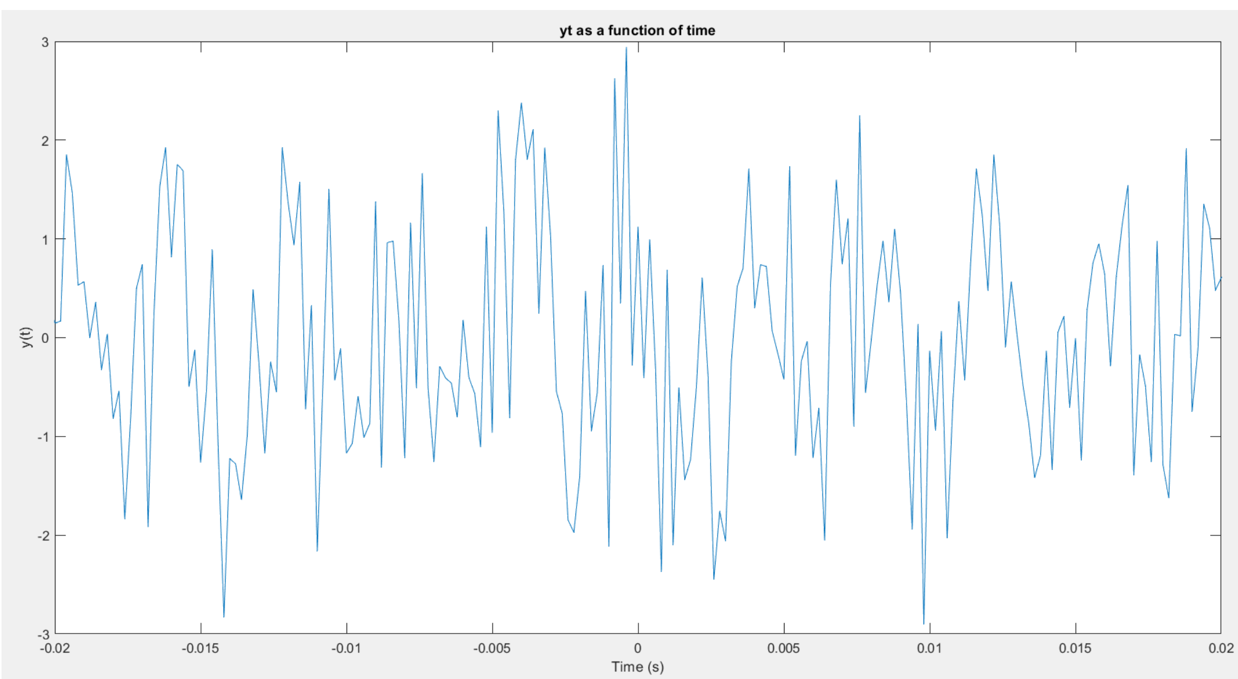
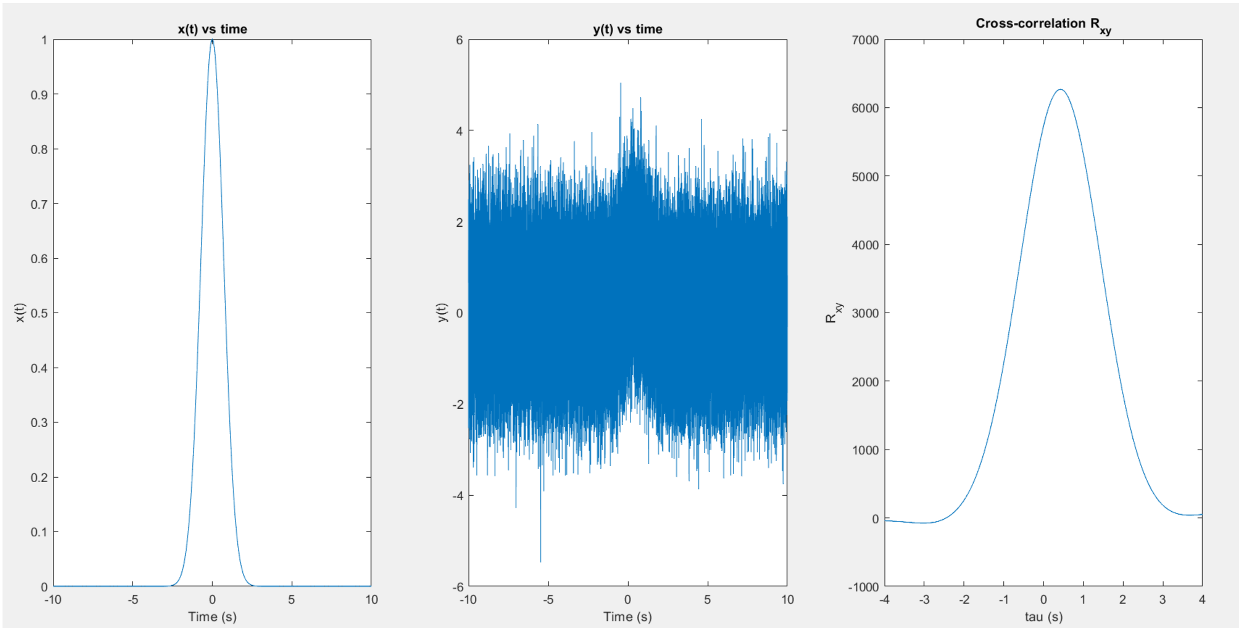


Figure 7. YT as a function of time domain

Numerical Experiment 3



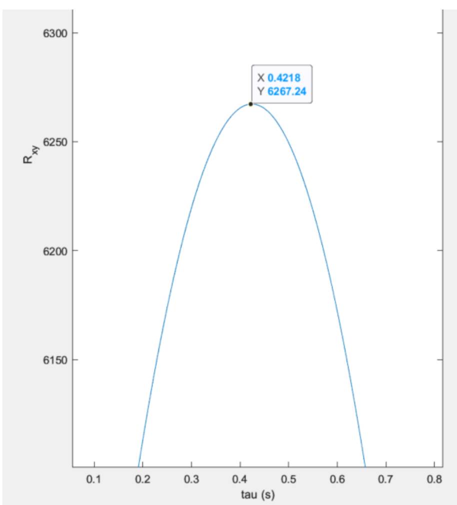


Figure 8. Peak value of cross-correlation

The delay of can be found by looking at the x value at which the peak of the cross-correlation occurs. Looking at figure 8, we can see that this occurs at x = 0.422. Plotting y(t) vs time fails because there is too much noise to clearly see when the peak is. We can always assume where the peak is, but it can be an error and it is not going to be consistent. This technique is just not accurate enough. Looking at the equation given, $y(t) = e^{-(t+\tau-T)^2} + w(t+\tau)$ we can see that when $\tau = T$, the equation becomes $y(t) = e^{-(t)^2} + w(T)$ which means it is the same as the equation for x(t). This means it is correlated with x. We cannot use y(t) to calculate the delay accurately because as we know from earlier, yt is correlated with itself at τ = 0. From the plots we have seen, we can see that the peak happens at 0 which means we will not see any delay.

Matlab Code

```
%This code sets up the time and frequency vectors for all the numerical
%experiments of Lab3
clear
format long e
tend = 10;
tbeg = -10;
N=100000;
tstep = (tend-tbeg)/N;
sampling_rate = 1/tstep;

%Time window
tt = tbeg:tstep:tend-tstep;

y1 = load('lab4_num_expt1')
y2 = load('lab4_num_expt2')


y3 = load('lab4_num_expt3')

% Experiment 1
maxlag = 100; % Change from 100 to 200 to 500 for part i, ii, iii respectively
%Autocorrelation of yt
Ry  = xcorr(y1.yt,y1.yt,maxlag);
%tau vector
tau_vec = -(maxlag*tstep):tstep:maxlag*tstep;
%Abs. PSD corresponding to yt
Sy = abs(fftshift(fft(fftshift(Ry))));
%define the frequency vector corresponding to tau_vec
Ntau = length(tau_vec);
%Nyquist sampling rate
fmax = sampling_rate/2;
fmin = -fmax;
fstep = (fmax-fmin)/Ntau;
%Frequency window
freq = fmin:fstep:fmax-fstep;

tiledlayout(1,2);

nexttile;
plot(tau_vec, Ry);
title("Autocorrelation of yt");
xlabel("tau (s)");
ylabel("Ry");


nexttile;
plot(freq, Sy);
title("PSD vs Frequency");
xlabel("Frequency (Hz)");
ylabel("PSD");


% Experiment 2
maxlag = 100; % Change from 100 to 200 to 20000 for part i, ii, iii respectively
%Autocorrelation of yt
Ry  = xcorr(y2.yt,y2.yt,maxlag);
%tau vector
tau_vec = -(maxlag*tstep):tstep:maxlag*tstep;
%Abs. PSD corresponding to yt
Sy = abs(fftshift(fft(fftshift(Ry))));
%define the frequency vector corresponding to tau_vec
Ntau = length(tau_vec);
%Nyquist sampling rate
fmax = sampling_rate/2;
fmin = -fmax;
fstep = (fmax-fmin)/Ntau;
%Frequency window
freq = fmin:fstep:fmax-fstep;
```

```
tiledlayout(1,2);
nexttile;
plot(tau_vec, Ry);
title("Autocorrelation of yt");
xlabel("tau (s)");
ylabel("Ry");

nexttile;
plot(freq, Sy);
title("PSD vs Frequency");
xlabel("Frequency (Hz)");
ylabel("PSD");

plot(tt, y2.yt);
title("yt as a function of time");
xlim([-100*tstep 100*tstep]);
xlabel("Time (s)");
ylabel("y(t)");

% Experiment 3
maxlag = 20000;
%Autocorrelation of yt
Rxy  = xcorr(y3.yt,y3.xt,maxlag);
%tau vector
tau_vec = -(maxlag*tstep):tstep:maxlag*tstep;
%Abs. PSD corresponding to yt
Sy = abs(fftshift(fft(fftshift(Rxy))));
%define the frequency vector corresponding to tau_vec
Ntau = length(tau_vec);
%Nyquist sampling rate
fmax = sampling_rate/2;
fmin = -fmax;
fstep = (fmax-fmin)/Ntau;
%Frequency window
freq = fmin:fstep:fmax-fstep;
```

```
freq = fmin:fstep:fmax-fstep;

tiledlayout(1,3);

nexttile;
plot(tt, y3.xt);
title("x(t) vs time");
xlabel("Time (s)");
ylabel("x(t)");

nexttile;
plot(tt, y3.yt);
title("y(t) vs time");
xlabel("Time (s)");
ylabel("y(t)");

nexttile;
plot(tau_vec, Rxy);
title("Cross-correlation R_{xy}");
xlabel("tau (s)");
ylabel("R_{xy}");
```