EE3TR4 – Lab 4 Report

Yiming Chen, 400230266

Ruiyi Deng, 400240387

cheny466@mcmaster.ca

dengr6@mcmaster.ca

April 10th, 2022

Experiment 1:

(i) The theoretical calculation process is shown below.

Autocorrelation $R_Y(m)$, calculation:

$$y(n) = h(n) * w(n)$$

$$h(n) = 2B \operatorname{sinc}(2B \cdot n)$$

$$= 500 \operatorname{sinc}(2 \cdot 250 n)$$

$$R_Y(m) = E\{y(n) y(n+m)\}$$

$$= \sum_k \sum_j h(k) h(j) E\{w(n-k) w(n+m-j)\}$$

$$= \sum_k h(k) h(k+m) \sigma_W^2$$

$$= \sum_k [500 \operatorname{sinc}(500k)][500 \operatorname{sinc}(500(k+m))] \sigma_W^2$$

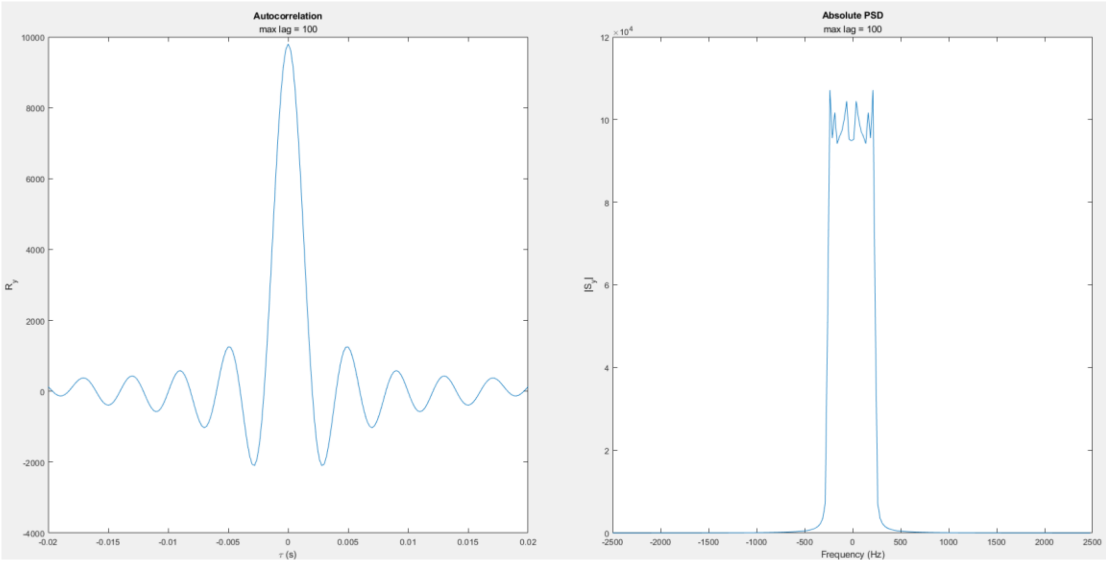
PSD calculation:

$$S_Y(f) = F\{R_Y(m)\}$$

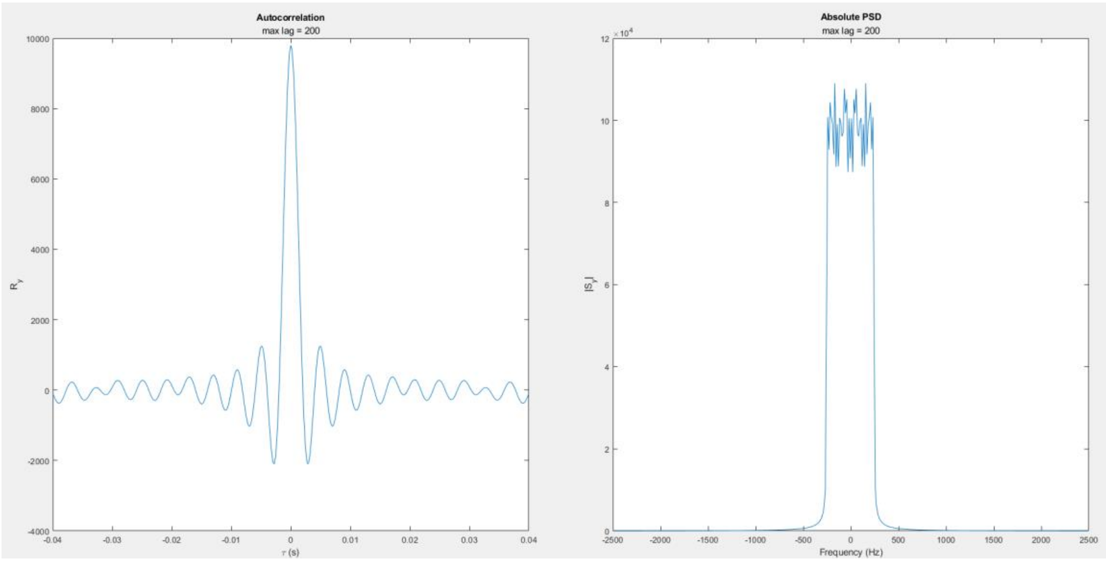
$$= k \cdot \operatorname{rect}(f/2 \cdot 250)$$

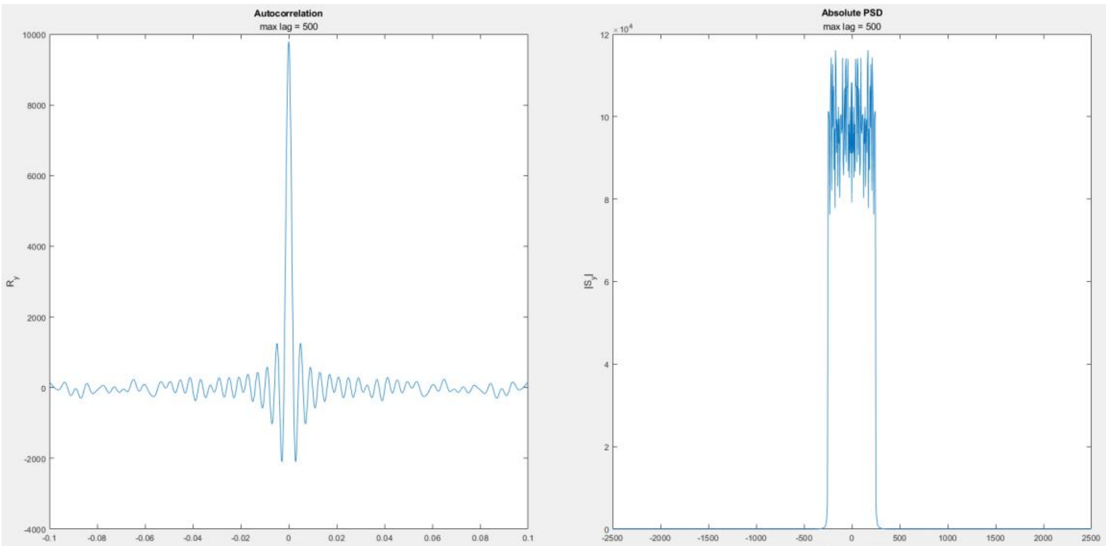
$$= k \cdot \operatorname{rect}(f/500)$$

From the calculation above we know that in the autocorrelation function the two sinc functions possess close frequency and the bandwidth of the PSD plot is 250 Hz. The matlab simulation is attached below. Comparing these two results we can say the theoretical results are very closed to the simulations ones. There exit some interference in simulation and we infer that the white noise signal implemented in the matlab is not perfectly ideal.



(ii) The simulation results with max lag at 200 and 500 are attached below.





From the results we know that increasing the max lag, the bandwidth stays the same. The resolution of the PSD increases which will help to deliver more message.

(iii) According to the three autocorrelation plots we can clearly find that the zeros appear at the interval of 2ms, which means the peaks repeats every 4ms. From this we can calculate the bandwidth of the filter is 250 Hz.

Experiment 2:

We can calculate the theoretical autocorrelation function of the output by:

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

$$R_y(\tau) = E\{y(t)y(t+\tau)\}$$

$$= E\left\{\left[A\sin\left(\underbrace{2\pi f_c t + \theta}_{x}\right) + w(t)\right]\cdot\left[A\sin\left(\underbrace{2\pi f_c(t+\tau) + \theta}_{y}\right) + w(t+\tau)\right]\right\}$$

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

$$= \frac{A^2}{2}\sin(2\pi f_c\tau) + 0 + 0 + \frac{N_0}{2}\delta(\tau)$$

$$= \frac{A^2}{2}\sin(2\pi f_c\tau) + \frac{N_0}{2}\delta(\tau)$$

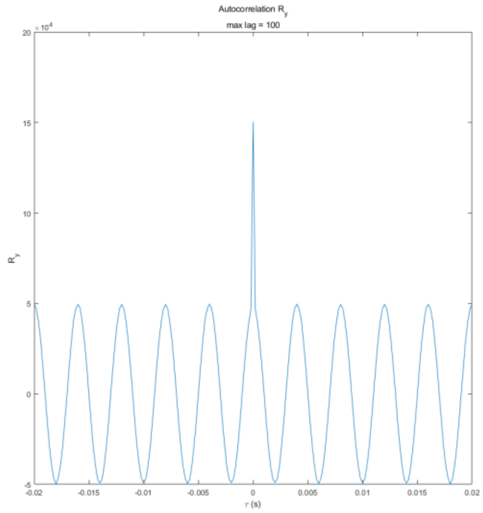
Since PSD is the Fourier transform of the autocorrelation function, so we can get the PSD by:

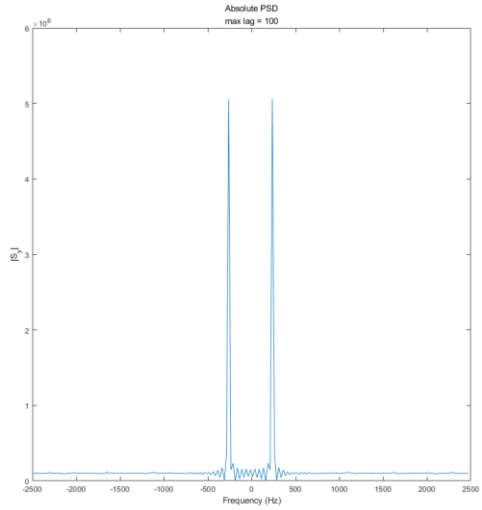
$$R_y(\tau) = \frac{A^2}{2}\sin(2\pi f_c\tau) + \frac{N_0}{2}\delta(\tau)$$

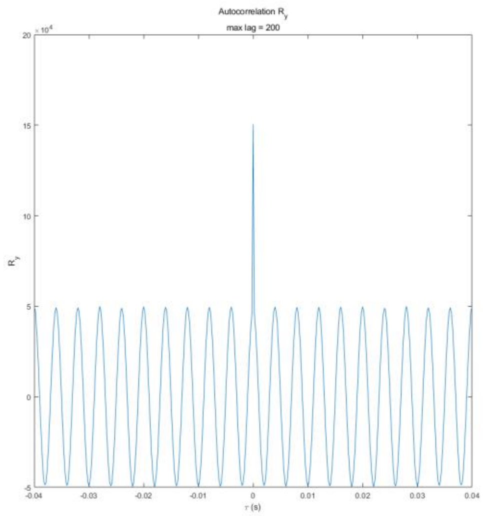
$$S_y(f) = \left|F\left\{\frac{A^2}{2}\sin(2\pi f_c\tau)\right\} + F\left\{\frac{N_0}{2}\delta(\tau)\right\}\right|$$

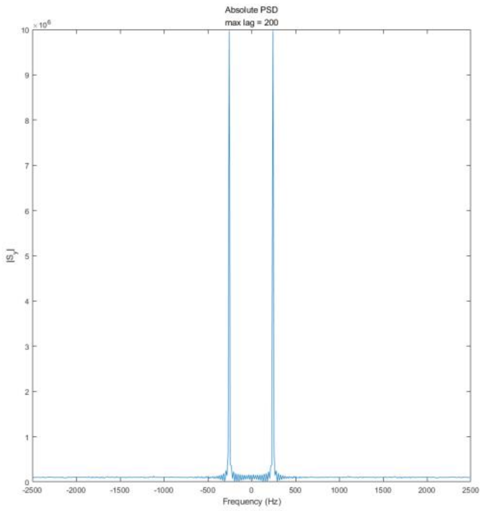
$$= \left|i\cdot\frac{A^2}{4}[\delta(f-f_c) - \delta(f+f_c)]\right| + \frac{N_0}{2}$$

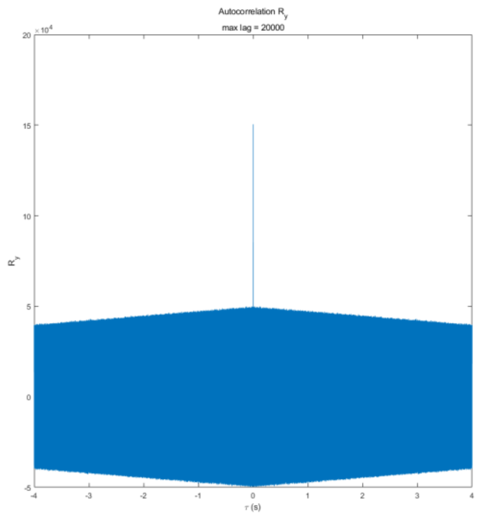
$$= \frac{A^2}{4}[\delta(f-f_c) - \delta(f+f_c)] + \frac{N_0}{2}$$

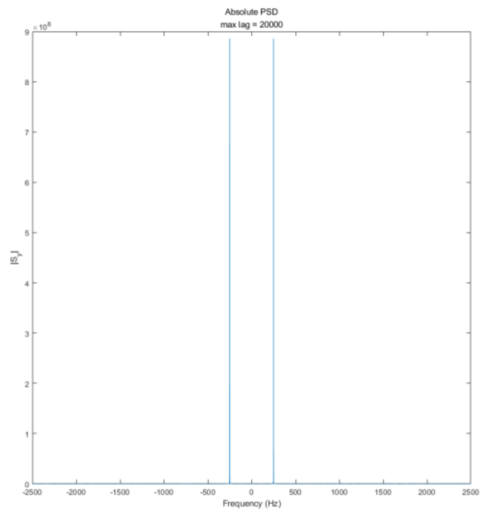










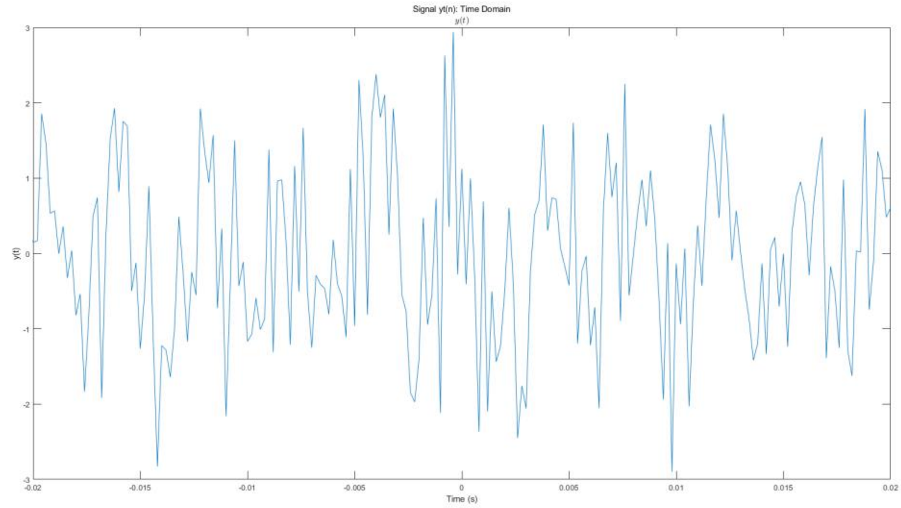


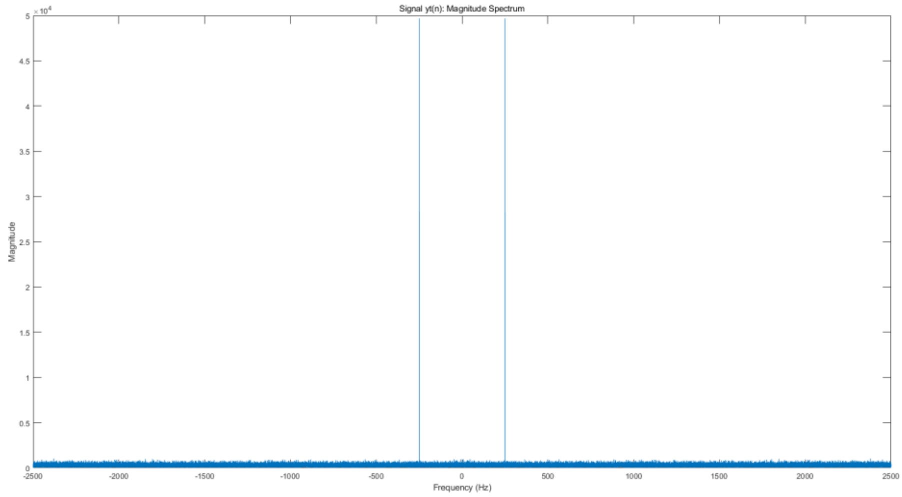
Maxlag	100	200	20000
f_c	237.35	244.16	249.84

The plots above are the results of maxlag = 100, 200, 20000 in the order of top to bottom. Compared with the theoretical results we get, we can see both the autocorrelation and PSD match between the theoretical results and the numerical results. Both autocorrelation functions result in sin functions with an impulse at $\tau = 0$. Also, both PSDs produce two impulses at what should be $-f_c$ and f_c with a floor caused by the noise.

i) We noticed that there is a peak at the zero lag in the autocorrelation plot and we think the reason of the presence of this peak is the impulse in the autocorrelation function due to the white noise.

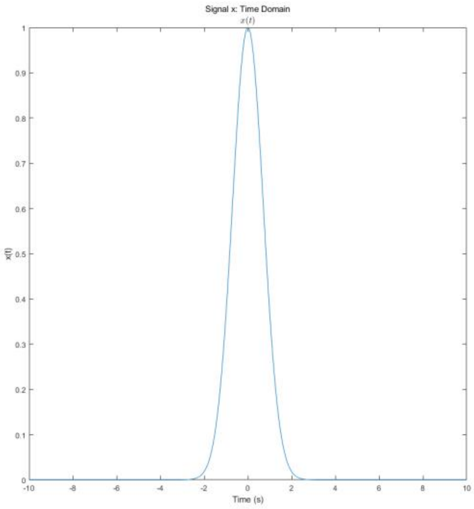
ii) From the plots above, we can find the impact of increasing the maxlag to 200 and 20000. We will get narrower peaks in the PSD plots as when we increase the maxlag, the frequency resolution of the PSD also increases. From the table above, we can see that the frequency estimate for f_c approaches 250Hz due to the increase in frequency resolution as the value of maxlag increases. As maxlag increases, the size of the τ vector increases. The size of the frequency vector is dependent on the size of the τ vector. As the range of the frequency vector does not change, the increase in the number of frequency samples in the vector due to an increase maxlag increases the resolution of the frequency vector.

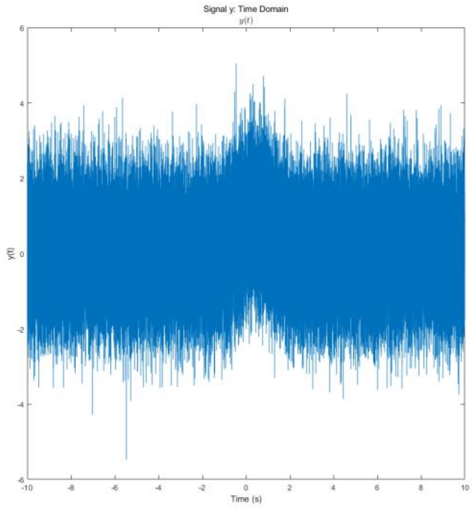


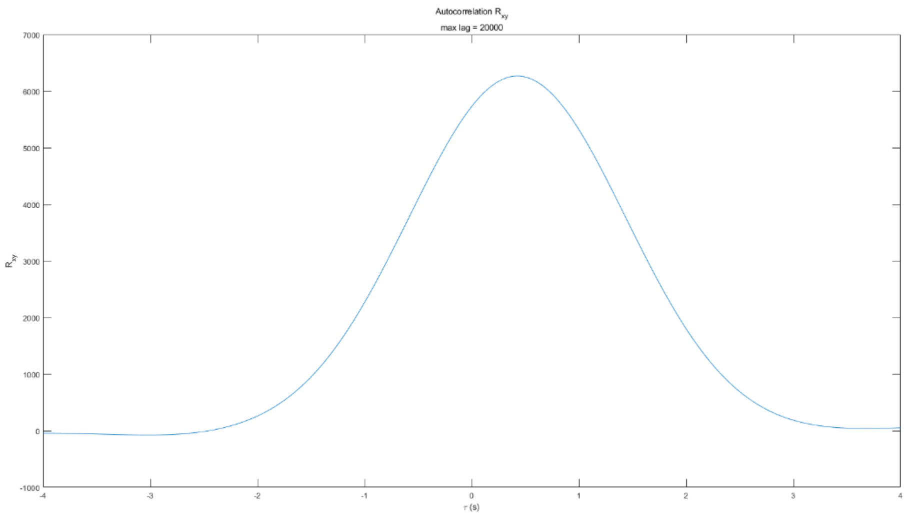


iii) Above is the time domain and the magnitude spectrum of signal *yt*. We can find that the sinusoidal signal is buried in noise. We can estimate the frequency f_c by taking the FFT of *yt.* We can clearly see from the magnitude spectrum plot that the output signal is around 250 Hz, which matches our estimation. We can estimate the frequency by looking at the output signal *yt* in the time domain without calculation the correlation.

Experiment 3:







Above are the plots of time domain of *x(t)* and *y(t)* and cross-correlation between *x* and *y*. From the plot of time domain, we think it is hard for us to determine what the delay is as the signal is almost completely buried in noise, which is hard to pinpoint the exact location of the peak of the delay. And we find plotting the cross-correlation between *x* and *y* to find the value of τ when the autocorrelation function reaches its maximum is an approach to determining the delay. From the plot above, we can see that the maximum value of the cross-correlation function occurs at τ = 0.0424s, which roughly matches the delay we found in the time domain plot. The autocorrelation of y (xcorr(y,y)) cannot be used to estimate the delay as the maximum value of the autocorrelation function occurs at τ = 0, which is not equal to the delay in the signal y.

Appendix: Matlab Code

```
format long e
tend = 10;
tbeg = -10;
N=100000;
tstep = (tend-tbeg)/N;
sampling_rate = 1/tstep;

tt = tbeg:tstep:tend-tstep;

%% Experiment 1
yt1 = load('lab4_num_expt1');
lag1 = [100 200 500];

for i = 1:3
    figure(i)
    tiledlayout(1,2);

    maxlag = lag1(i);
    %Autocorrelation of yt
    Ry  = xcorr(yt1.yt,yt1.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;

    nexttile;
    plot(tau_vec, Ry);
    title("Autocorrelation");
    subtitle("max lag = " + maxlag);
    xlabel("\tau (s)");
    ylabel("R_y");

    nexttile;
    plot(freq, Sy);
    title("Absolute PSD");
    subtitle("max lag = " + maxlag);
    xlabel("Frequency (Hz)");
    ylabel("|S_y|");

end
```

```matlab
%% Experiment 2
yt2 = load('lab4_num_expt2');
lag2 = [100 200 20000];

for i = 4:6
    figure(i);
    tiledlayout(1,2);

    maxlag = lag2(i-3);
    %Autocorrelation of yt
    Ry  = xcorr(yt2.yt,yt2.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;

    nexttile;
    plot(tau_vec, Ry);
    title("Autocorrelation R_y");
    subtitle("max lag = " + maxlag);
    xlabel("\tau (s)");
    ylabel("R_y");

    nexttile;
    plot(freq, Sy);
    title("Absolute PSD");
    subtitle("max lag = " + maxlag);
    xlabel("Frequency (Hz)");
    ylabel("|S_y|");


end

% Signal in time domain
figure(7);
plot(tt, yt2.yt);
```

```
xlim([-100*tstep 100*tstep]);
title("Signal yt(n): Time Domain");
subtitle("$y(t)$", 'interpreter', 'latex')
xlabel("Time (s)");
ylabel("y(t)");

% Signal in frequency domain
figure(8);

Nyt = length(yt2.yt);
% Nyquist sampling rate
fmax = sampling_rate/2;
fmin = -fmax;
fstep = (fmax-fmin)/Nyt;
% Frequency window
freq = fmin:fstep:fmax-fstep;

plot(freq, abs(fftshift(fft(fftshift(yt2.yt)))));
title("Signal yt(n): Magnitude Spectrum");
xlabel("Frequency (Hz)");
ylabel("Magnitude");
```

```
%% Experiment 3
value3 = load('lab4_num_expt3');
xt3 = value3.xt;
yt3 = value3.yt;

figure(9);
tiledlayout(1,2);

nexttile;
plot(tt, xt3);
title("Signal x: Time Domain");
subtitle("$x(t)$", 'interpreter', 'latex');
xlabel("Time (s)");
ylabel("x(t)");

nexttile;
plot(tt, yt3);
title("Signal y: Time Domain");
subtitle("$y(t)$", 'interpreter', 'latex');
xlabel("Time (s)");
ylabel("y(t)");
```

```
maxlag = 20000;
R_xy = xcorr(yt3,xt3,maxlag);
tau_vec = -(maxlag*tstep):tstep:maxlag*tstep;
%Abs. PSD corresponding to yt
S_xy = abs(fftshift(fft(fftshift(R_xy))));
%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;

fig = figure(10);
plot(tau_vec, R_xy);
title("Autocorrelation R_{xy}");
subtitle("max lag = " + maxlag)
xlabel("\tau (s)");
ylabel("R_{xy}");
```