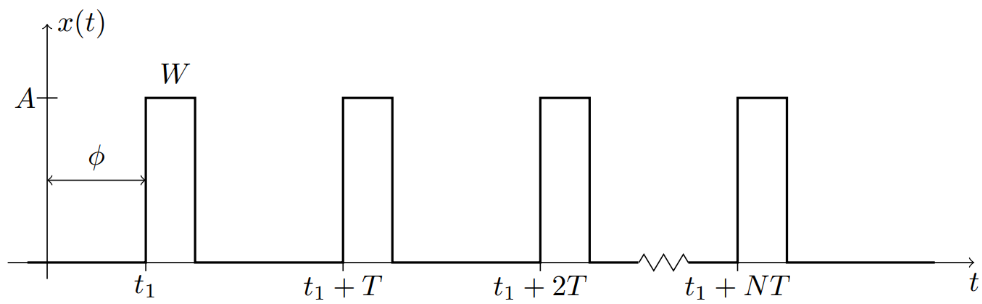
# Homework 5

**Table of Contents**



## [✔] ADSI Problem 4.2: Autocorrelation functions from plot

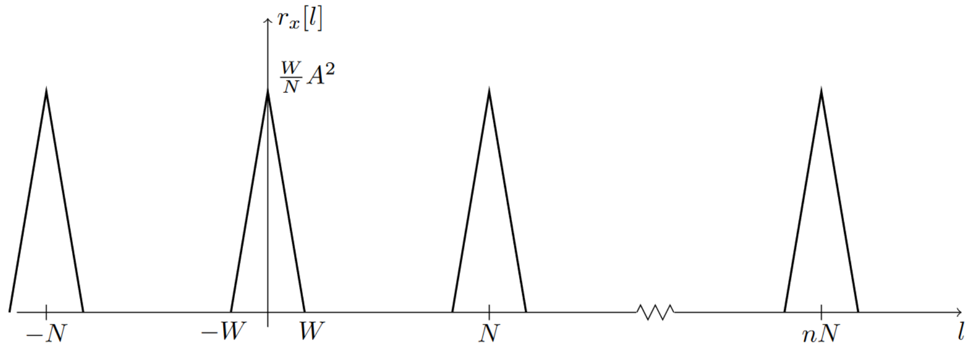
The outcome of a random process is a pulse train with period $T$ and pulse width $W$ as shown in the graph. Let the phase $\phi$ be uniformly distributed between 0 and $T$ so that the random process is stationary.




### 1) Sketch and explain what the autocorrelation looks like

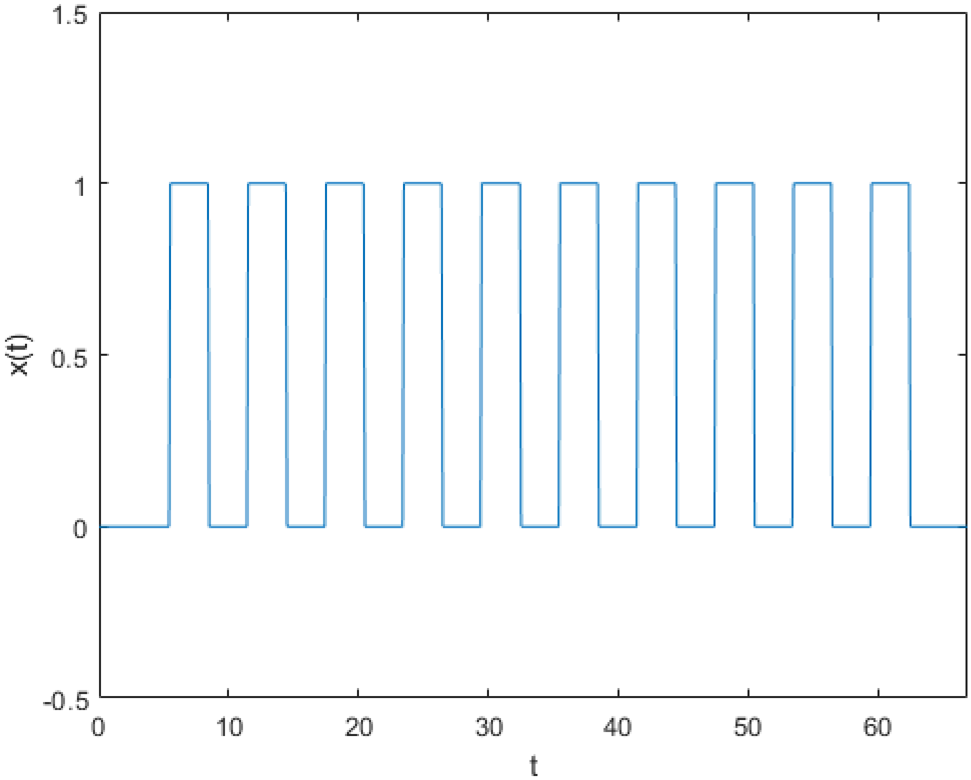
Determine, using only simple arguments and drawings, what the autocorrelation function $R_{XX}(\tau)$ will look like.

First of all, we note the random process is periodic. This periodicity is retained in the autocorrelation. The autocorrelation for a stationary function is defined by $r_l[l] = E(x[n]x[n-l])$. For $l = 0$ we therefore get $r_l[0] = E(x[n]x[n])$. When we multiply the signal with itself we see that throughout one period we get $A^2$ in a fraction $\frac{W}{N}$ of the period and 0 in the remaining $1-\frac{W}{N}$ of the period. Therefore $r_l[0] = \frac{W}{N}A^2$. If $l = nN$ where $n \in \mathbb{Z}$ we get the same result as for $l = 0$. When we increase (or decrease) $l$ away from 0, the overlap between $x[n]$ and $x[n-l]$ linearly decreases until the overlap is zero at $|l| = W$. The overlap between $x[n]$ and $x[n-l]$ remains at zero until $l = N-W$ (or $l = -N+W$). This repeats each time the lag is increased by N. The autocorrelation function therefore looks like an infinite series of triangular spikes.



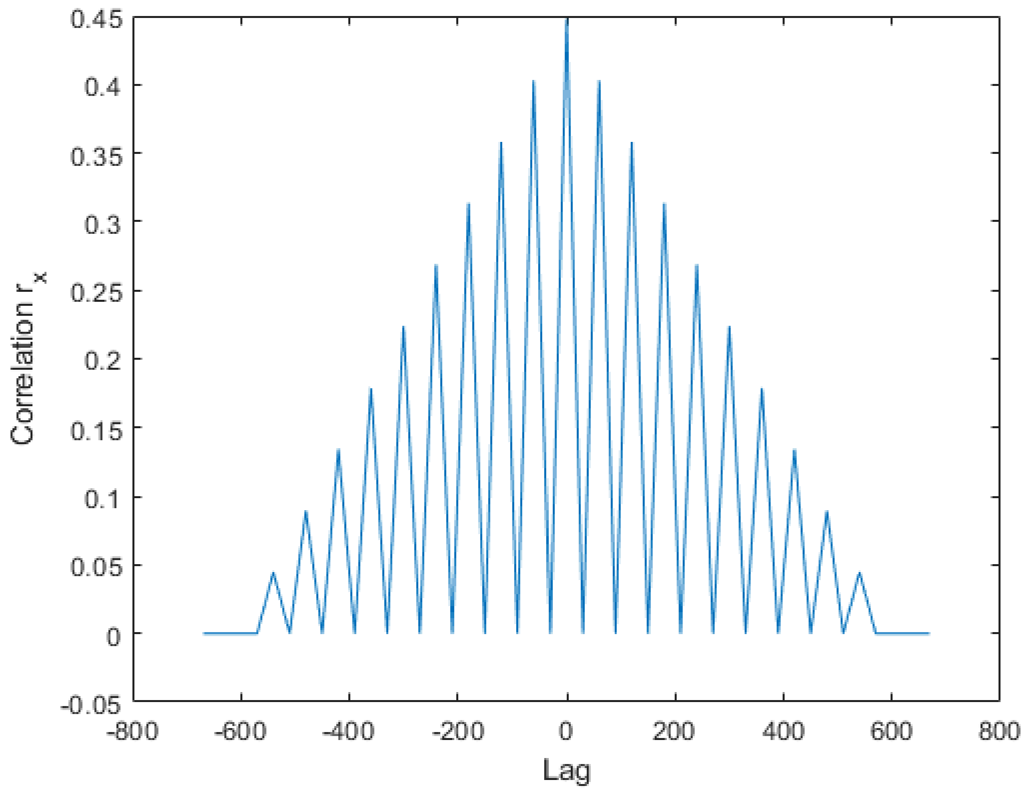

## 2) Use computer simulations to verify the above result

```
A = 1;
W = 3;  % Pulse width
T = 6; % Period width
N = 10;

% Random number from a uniform distribution in the open interval (a, b)
% a = 0; b = T; random_number = (b-a)*rand + a
phi = T*rand(1);

maxT = phi+W/2 + T*N;

t=0:.1:maxT;   % Time vector
d= phi+W/2:T:phi+W/2+T*N-1; % Delay vector
y = pulstran(t,d,'rectpuls', W);
plot(t,y)

ylabel('x(t)');
xlabel('t');
ylim([-0.5, A+0.5]);
xlim([0, max(t)]);
```




Autocorrelation is symmetrical around zero.

```
[r_xx, lags] = xcorr(y, 'biased');
plot(lags, r_xx);
ylabel('Correlation r_x');
xlabel('Lag')
```




## [✔] ADSI Problem 4.4: Autocorrelation of complex signals

The autocorrelation and crosscorrelation function for complex signals are given by

$$r_x(l) = E[x(n)x^*(n-l)], \qquad r_{yx}(l) = E[y(n)x^*(n-l)]$$

Consider a complex sinusoid given by $x(n) = Ae^{j(\omega n+\phi)}$ where $A$ and $\omega$ are real constants and $\phi$ is a random variable with $\phi \sim U(0, 2\pi)$.

### 1) Compute the analytic expression for the autocorrelation of the complex sinusoid

1. Compute the analytic expression for the autocorrelation of the complex sinusoid.

The autocorrelation function for complex signals can be computed as:

$r_{xx}(\ell) = E[x(n)x^*(n-\ell)]$

Plugging the given complex sinusiod into the formula, we get:

$r_{xx}(\ell) = E[A\, e^{j(\omega n+\phi)} A\, e^{-j(\omega(n-\ell)+\phi)}]$

Since $A$ is a constant, we can move it outside the expected value:

$$r_{xx}(\ell) = A^2 E[\, e^{j(\omega n+\phi)}\, e^{-j(\omega(n-\ell)+\phi)}]$$

$$r_{xx}(\ell) = A^2 E[\, e^{j\omega n+j\phi}\, e^{-j\omega n+j\omega\ell-j\phi}]$$

$$r_{xx}(\ell) = A^2 E[\, e^{j\omega n+j\phi+(-j\omega n+j\omega\ell-j\phi)}\, ]$$

$$r_{xx}(\ell) = A^2 E[\, e^{j\omega n+j\phi-j\omega n+j\omega\ell-j\phi}\, ]$$

$$r_{xx}(\ell) = A^2 E[\, e^{j\omega\ell}\, ]$$

We know that $E[e^{j\omega\ell}] = e^{j\omega\ell}$ because the expected value of a constant is just the constant itself. Notice that $\phi$ is no longer in the expression $e^{j\omega\ell}$. Therefore, the autocorrelation of the complex sinusoid is:

$$r_{xx}(\ell) = A^2\, e^{j\omega\ell}$$

## 2) How are rx(l) and rx(-l) related for a complex signal?

2. How are $r_x(l)$ and $r_x(-l)$ related when $x(n)$ is a complex-valued signal?

In 1) we found an analytical expression for $r_{xx}(\ell) = A^2\, e^{j\omega\ell}$

Let us compute the autocorrelation for $-\ell$:

$$r_{xx}(-\ell) = E[x(n)x^*(n+\ell)]$$

Suppose $m = n + \ell$. Then we can write $n = m - \ell$. Let us substitute all $n$ with $m$ in the above expression:

$$r_{xx}(-\ell) = E[x(m-\ell)x^*(m)]$$

Computing the complex conjucate of the expectation we get:

$$r_{xx}(-\ell) = E[x^*(m-\ell)x(m)]^*$$

$$r_{xx}(-\ell) = E[x(m)x^*(m-\ell)]^*$$

Since $r^*_{xx}(\ell) = E[x(m)x^*(m-\ell)]^*$, we know that:

$$r_{xx}(-\ell) = r^*_{xx}(\ell)$$

## 3) Compute the autocorrelation of a real signal

3. Use the above result to compute the autocorrelation of the real signal $x(n) = A\cos(\omega n + \phi)$.

We can use the result from 1)

$r_{xx}(\ell) = A^2 e^{j\omega\ell}$

Since the result from 1) uses Euler, we need to convert the signal to complex exponential.

We use the relation $\cos(\theta) = \frac{1}{2}(e^{j\theta} + e^{-j\theta})$

$$x(n) = A\cos(\omega n + \phi)$$

$$x(n) = \frac{A}{2}\left(e^{j(\omega n+\phi)} + e^{-j(\omega n+\phi)}\right)$$

$$x(n) = \frac{A}{2}e^{j(\omega n+\phi)} + \frac{A}{2}e^{-j(\omega n+\phi)}$$

In 1) we found that the autocorrelation of a complex sinusoid $x(n) = A\, e^{j(\omega n+\phi)}$ is $r_{xx}(\ell) = A^2 e^{j\omega\ell}$

Therefore, the autocorrelation of the real signal is:

$$r_{xx}(\ell) = \left(\frac{A}{2}\right)^2 e^{j\omega\ell} + \left(\frac{A}{2}\right)^2 e^{-j\omega\ell}$$

$$r_{xx}(\ell) = \frac{A^2}{4}e^{j\omega\ell} + \frac{A^2}{4}e^{-j\omega\ell}$$

$$r_{xx}(\ell) = \frac{A^2}{4}(e^{j\omega\ell} + e^{-j\omega\ell})$$

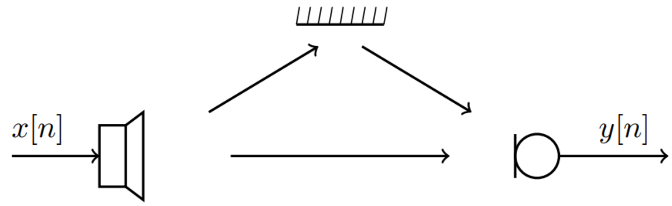
$$r_{xx}(\ell) = \frac{A^2}{2}\frac{1}{2}(e^{j\omega\ell} + e^{-j\omega\ell})$$

Using the relation $\cos(\theta) = \frac{1}{2}(e^{j\theta} + e^{-j\theta})$ we can rewrite the autocorrelation to:

$$r_{xx}(\ell) = \frac{A^2}{2}\cos(\omega\ell)$$

## [✔] ADSI Problem 4.5: Autocorrelation of distorted signals

An audio signal is sent across a room from a loudspeaker to a microphone as shown in the sketch below and is distorted by a reflection from a surface. The audio signal played in the loudspeaker $x[n]$ is almost white noise, i.e. the autocorrelation function contains only a few non-zero values, all at lags close to zero.




## 1) Derive autocorrelation function of the output signal

1. Derive and sketch the autocorrelation function of the $y[n]$ signal recorded at the microphone. Account for any assumptions and simplifications used in the derivation.

From the figure, we observed that the signal at microphone $y(n)$ is a combination of two signals.

The bottom arrow describes the signal from the loudspeaker that reaches the microphone with no distortion and no delays.

The two top arrows describes the signal that is distorted by a reflection. This means that the original signal $x(n)$ will be attenuated and delayed.

Therefore, we can describe the signal at the microphone as

$$y(n) = x(n) + \alpha x(n - D)$$

where

- $\alpha < 1$ is the attenuation caused by the reflection
- $D$ is the delay of the original signal

The autocorrelation of $y(n)$ can be derived:

$$r_{yy}(\ell) = E[y(n)y(n - \ell)]$$

$$r_{yy}(\ell) = E[(x(n) + \alpha x(n - D))(x(n - \ell) + \alpha x(n - D - \ell))]$$

Multiple each terms

$$r_{yy}(\ell) = E[x(n)x(n - \ell) + \alpha x(n)x(n - D - \ell) + \alpha x(n - D)x(n - \ell) + \alpha^2 x(n - D)x(n - D - \ell)]$$

Seperate expectation terms

$$r_{yy}(\ell) = E[x(n)x(n - \ell)] + E[\alpha x(n)x(n - D - \ell)] + E[\alpha x(n - D)x(n - \ell)] + E[\alpha^2 x(n - D)x(n - D - \ell)]$$

Move constants out of the expecation:

$$r_{yy}(\ell) = E[x(n)x(n-\ell)] + \alpha E[x(n)x(n-D-\ell)] + \alpha E[x(n-D)x(n-\ell)] + \alpha^2 E[x(n-D)x(n-D-\ell)]$$

We will use the relation $r_{xx}(\ell) = E[x(n)x(n-\ell)]$ to rewrite the expression above.

The expectation in the first term and the last term are trivial to rewrite:

$$r_{yy}(\ell) = r_{xx}(\ell) + \alpha E[x(n)x(n-D-\ell)] + \alpha E[x(n-D)x(n-\ell)] + \alpha^2 r_{xx}(\ell)$$

We know that $r_{xx}(\ell - D) = E[x(n)x(n-\ell-D)]$ so we want rewrite the second term:

$$r_{yy}(\ell) = r_{xx}(\ell) + \alpha r_{xx}(\ell - D) + \alpha E[x(n-D)x(n-\ell)] + \alpha^2 r_{xx}(\ell)$$

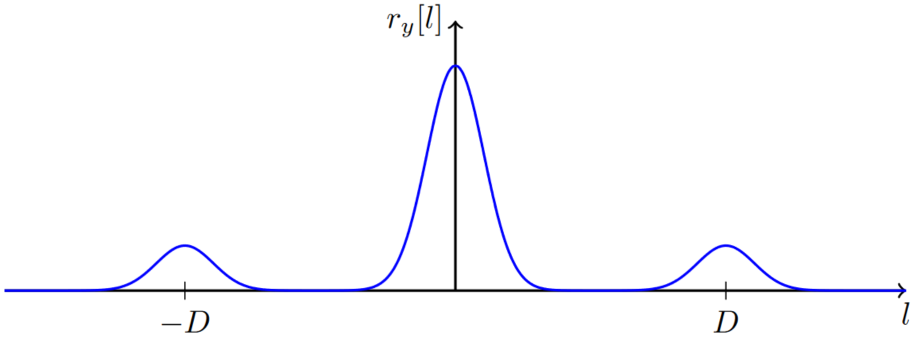
Let $m = n - D$ then $n = m + D$. Then $E[x(m)x(m-\ell+D)]$. Thus, the third term can rewritten to:

$$r_{yy}(\ell) = r_{xx}(\ell) + \alpha r_{xx}(\ell - D) + \alpha r_{xx}(\ell + D) + \alpha^2 r_{xx}(\ell)$$

To make it easier to sketch the autocorrelation $r_{yy}(\ell)$, we can rewrite it the expression as:

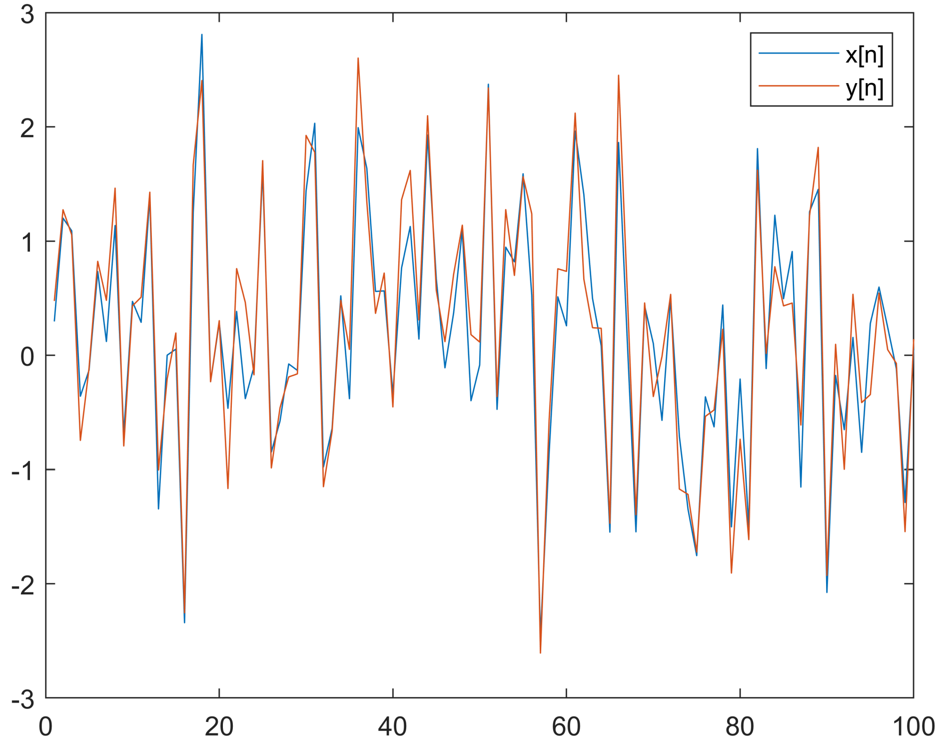
$$r_{yy}(\ell) = (1+\alpha^2) r_{xx}(\ell) + \alpha r_{xx}(\ell - D) + \alpha r_{xx}(\ell + D)$$

From this expression, we know that there are three peaks because each autocorrelation $r_{xx}$ corresponds to a peak. One peak is at lag zero, another peak at lag $-D$ and the last peak at $D$. Therefore,

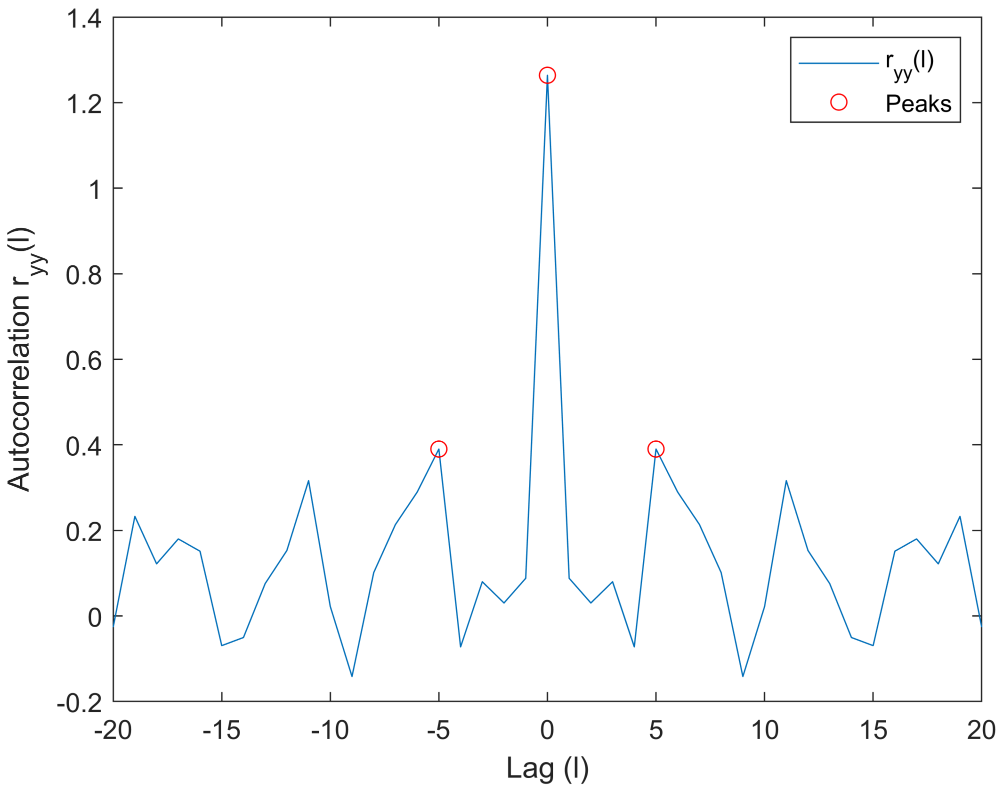



Let us check if we can see the same patterns if we simulate it in MATLAB:

```
alpha = 0.3;
D = 5;
N = 100;

n = [1:N];
x = wgn(N,1,0);
y = x + alpha*circshift(x, D);
plot(n, x, n, y)
legend('x[n]', 'y[n]')
```




```
[r_yy, lags] = xcorr(y, 20, 'biased');
plot(lags, r_yy);
hold on;
plot(0, r_yy(find(lags == 0)), 'ro');
plot(D, r_yy(find(lags == D)), 'ro');
plot(-D, r_yy(find(lags == -D)), 'ro');
legend('r_{yy}(l)', 'Peaks')
ylabel('Autocorrelation r_{yy}(l)');
xlabel('Lag (l)')
hold off;
```




**2) Is possible to remove the distortion from the output signal?**

2. Is it possible to remove the distortion from the $y[n]$ signal?

From our sketch, we can estimate the delay and attenuation by the position of the peaks and the amplitude ratio.

Then we create a FIR filter which substract a delayed and attenuated copy of $y[n]$:

$z[n] = y[n] - \alpha y[n - D]$

$z[n] = x[n] + \alpha x[n - D] - \alpha(x[n - D] + \alpha x[(n - D) - D])$

$z[n] = x[n] + \alpha x[n - D] - \alpha(x[n - D] + \alpha x[n - 2D])$

$z[n] = x[n] + \alpha x[n - D] - \alpha x[n - D] - \alpha^2 x[n - 2D]$

$z[n] = x[n] - \alpha^2 x[n - 2D]$

The result is a new signal $z[n]$ where the distortion is pushed to $2D$ and the attenuation factor is squared.

Since $\alpha < 1$, the squared attenuation $\alpha^2$ is much smaller than 1. This means the distortion component is much smaller than before.

Another approach is as follows:

The distoration corresponds to a FIR filter with the system function $H(z) = 1 + \alpha z^{-D}$. In principle, this can be removed using an IIR filter with the inverse system function.

## ADSI Problem 4.6: System identification

Consider a signal $x[n]$ be the output of an ARMA(2,2) process with unit-variance, white Gaussian noise as input.

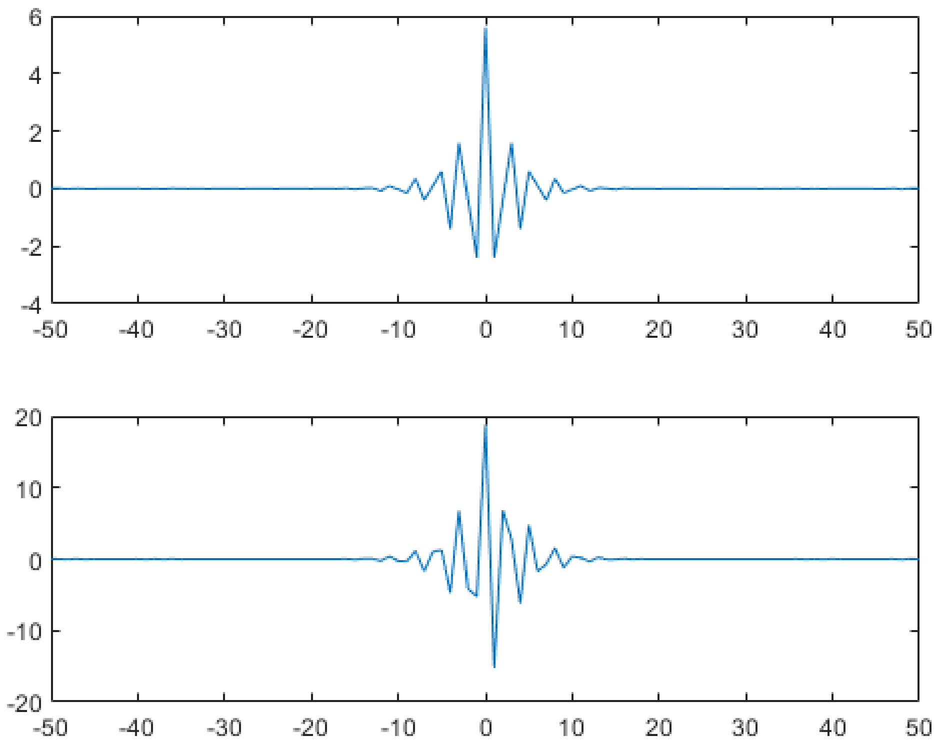
$$x[n] = 2w[n] + 1w[n-1] - x[n-1] - \frac{1}{2}x[n-2]$$

The signal is passed through a FIR filter with impulse response $h[n] = \{3,\ -1,\ 1\}$ to give $y[n]$.

### 1) Simulate the autocorrelation and cross-correlation in MATLAB

1. Simulate $r_{xx}[l]$ and $r_{yx}[l]$ in MATLAB.

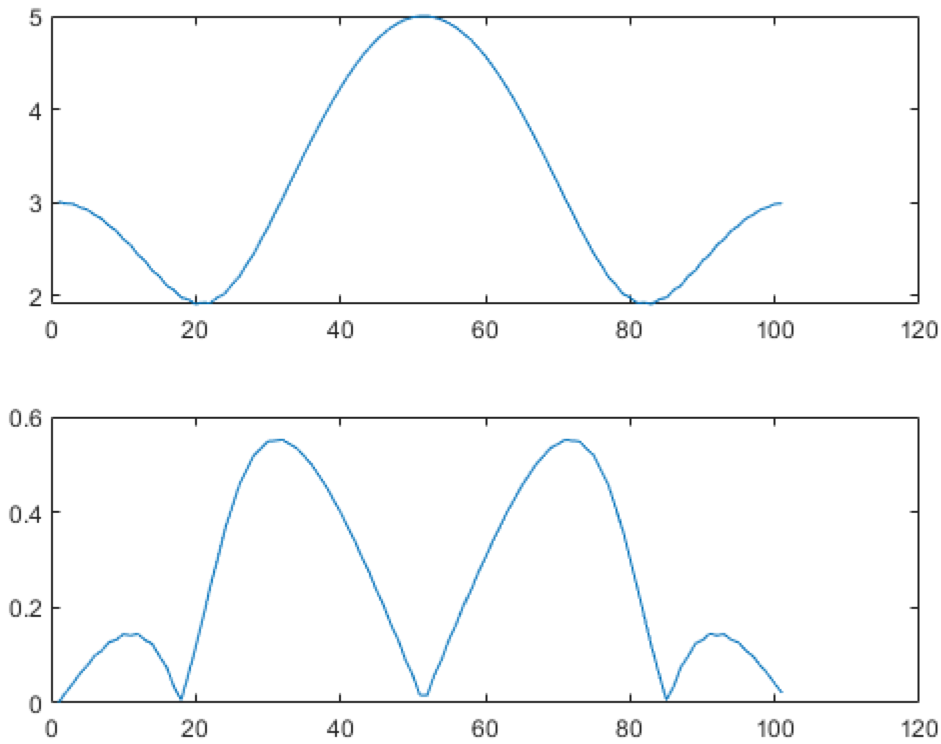
```
w=randn(1e6,1);
x=filter([2 1],[1 1 0.5],w);
y=filter([3 -1 1],1,x);

[rxx,lagsrxx]=xcorr(x,50,'biased');
[ryx,lagsryx]=xcorr(y,x,50,'biased');


figure(1)
subplot(211)
plot(lagsrxx,rxx)

subplot(212)
plot(lagsryx,ryx)
```



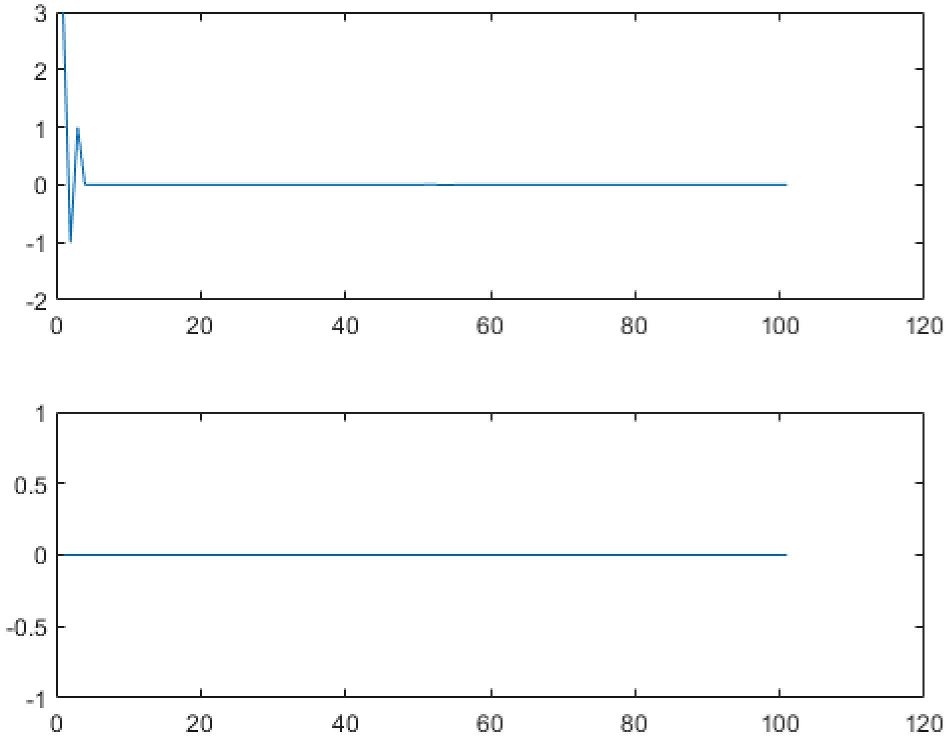

## 2) Compute the transfer function and impulse response from the autocorrelation

2. Compute $H(\omega)$ and $h[n]$ from $r_{xx}[l]$ and $r_{yx}[l]$ and compare with the analytical results.

```
%%
Rxx=fft(rxx);
Ryx=fft(ryx);

H=Ryx./Rxx;

figure(2)
subplot(211)
plot(abs(H)) % todo: add a frequency axis and labels.
subplot(212)
plot(abs(angle(H)))
```

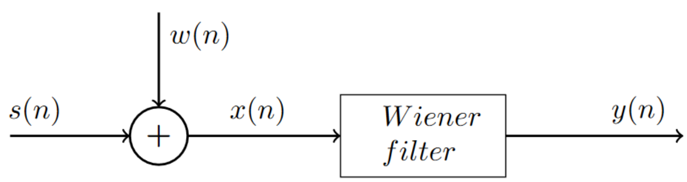



```
h=ifft(H);
figure(3)
subplot(211)
plot(real(h))

subplot(212)
plot(imag(h))
```




## Exam 2012 Problem 4: Wiener filter for recovering corrupted signal

A periodic signal $s(n)$ is corrupted by uncorrelated, additive noise $w(n)$ as shown in the figure.



It is desired to recover the signal by a Wiener filter. The autocorrelation of the noise is given $r_w(l) = e^{-0.2\sqrt{|l|}}$. The first values of the autocorrelation of the signal is given in the table below.

| $\lvert l \rvert$ | $r_s(l)$ |
|---|---|
| 0 | 1 |
| 1 | -0.4 |
| 2 | 0.2 |
| 3 | 0.1 |

## 1) Design a 3 tap Wiener filter to recover s(n)

## 2) Calculate the minimum mean square error

## 3) Discuss another Wiener filter with 2x noise amplitude

Consider a second, identical Wiener filtering problem, except that the noise amplitude is twice as big.

3. Discuss whether this second system will have the same minimum mean square error as the first system.