→ KALPIT BORKAR 200070029

→ NAVNEET 200070048

→ SANJHI PRIYA 20D070070

→ YUVRAJ SINGH 200070093

1. Given,

$R_{xx}(\tau)$ for $\tau = \ldots -2, -1, 0, 1, 2 \ldots\ldots$, and

$\hat{X}(n+1) = a\,X(n)$ is linear estimate for $X(n+1)$

For zero mean WSS sequence $X(n)$, we know that

$E[X_n] = 0 \quad \forall n \in Z$ (Z as its a discrete case).

$R_X[n_1, n_2] = E[X[n_1]\,X[n_2]] = R_X[n_1 - n_2]$ (i).

We have to find the minimum mean square error linear estimate for X_{n+1}, or minimizing.

$E[(X(n+1) - \hat{X}(n+1))^2]$

$\Rightarrow MSE(a) = E[(X(n+1) - a\,X(n))^2]$

$= E[(X(n+1))^2 + a^2 (X(n))^2 - 2a\,X(n+1)X(n)]$

$= E[X(n+1).X(n+1)] + a^2 E[X(n)\,X(n)] - 2a\,E[X(n+1)(X(n))]$

By (i).

$MSE(a) = R_{xx}(0) + a^2 R_{xx}(0) - 2a\,R_{xx}(1)$.

$MSE'(a) = 2a\,R_{xx}(0) - 2\,R_{xx}(1)$.

For minima,

$MSE'(a) = 0$, we get $a = \dfrac{R_{xx}(1)}{R_{xx}(0)}$.

As this is a quadratic equation, maxima will be at $a \to \infty$.

So our, $a = \dfrac{R_{xx}(1)}{R_{xx}(0)}$, is the minima.

$MMSE = R_{xx}(0) + \dfrac{R_{xx}^2(1)}{R_{xx}(0)} - \dfrac{2R_{xx}^2(1)}{R_{xx}(0)} = R_{xx}(0) - \dfrac{R_{xx}^2(1)}{R_{xx}(0)}$.

MMSE linear estimate for $X(n+1)$ is

$$\hat{X}(n+1) = \frac{R_{xx}(1)}{R_{xx}(0)}\,X(n)$$

Question 2:

Given:

$x(t)$ is real valued signal.

$$S_{xx}(\omega) = 3\left[U(\omega-9000) - U(\omega-11000)\right] + 400\,\delta(\omega-10000)$$

where $U(\omega)$ is the step function.

Sol:

$x(t)$ is a real valued signal $\Rightarrow$ graph in frequency domain is symmetrical. — (1)

$S_{xx}(\omega)$ is the Fourier transform of $R_{xx}(\tau)$.

$\therefore$ $R_{xx}(\tau)$ can be obtained by taking inverse Fourier transform of $S_{xx}(\omega)$.

$$\therefore R_{xx}(\tau) = \frac{1 \cdot 2}{2\pi} \int_{-\infty}^{\infty} S_{xx}(\tau)\, e^{j\omega\tau}\, d\omega$$

(factor of 2 from (1)).

$$\therefore R_{xx}(\tau) = \frac{1}{\pi}\left\{\int_{9000}^{11000} 3 \cdot e^{j\omega\tau} d\omega + \delta(\omega-10000) \cdot 400\right\}$$

$$= \frac{1}{\pi}\left\{3 \left[\frac{e^{j\omega\tau}}{j\tau}\right]_{9000}^{11000} + 400\right\}$$

$$= \frac{1}{\pi}\left[\frac{3}{j\tau}\left[e^{j11000\tau} - e^{j9000\tau}\right] + 400\right]$$

$$\therefore R_{xx}(\tau) = \frac{3}{\pi j \tau}\left[e^{j11000\tau} - e^{j9000\tau}\right] + 400.$$

to get $E[x^2(t)]$, we substitute $\tau = 0$ in R_{xx},

$$E[x^2(t)] = R_{xx}(0) = \frac{6400}{\pi}$$

Also, power $= R_{xx}(0) = \frac{6400}{\pi}$.

Assuming ergodicity,

$$E[x(t)] = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)\,dt \quad \text{— ①}$$

let $$x_1(\omega) = \lim_{T\to\infty} \int_{-T}^{T} x(t) e^{-j\omega t}\,dt \quad (F.T.)$$

$$\therefore x_1(0) = \lim_{T\to\infty} \int_{-T}^{T} x(t)\,dt \quad \text{— ②}$$

from eq ① & ②,

$$E[x(t)] = \lim_{T\to\infty} \frac{1}{2T} x_1(0)$$

We know, $$S_{xx} = \lim_{T\to\infty} \int_{-T}^{T} R_{xx}(t) e^{-j\omega t}\,dt$$

$$= \lim_{T\to\infty} \int_{-T}^{T} \cancel{x[x} \; E[x^2] e^{-j\omega t}\,dt$$

$$S_{xx}(0) = \lim_{T\to\infty} \int_{-T}^{T} E[x^2]\,dt \quad \text{— ③}$$

Similarly to ③, & ② $E[x^2(t)] = (x_1(0))^2$ [assuming ergodicity]

$$\therefore S_{xx}(0) = (x_1(0))^2 = 0. \quad (\text{from eq of } S_{xx}).$$

$$\therefore \boxed{E[x(t)] = 0}$$

Q3 a) The $E(Q_n]$ is varying as n changes so we can't Say that for part (a) it's stationary.

For part (b) also $E[Q_n]$ is Varying as n vary, so it's also not stationary

For part (c) $E[Q_n]$ is ~~very~~ not varying much, we can consider it to be stationary,

Since, for part a, b process is not WSS, we will not talk about ergodicity,

For part c, ensemble average = time average So, part c is ergodic also localhost:8888/notebooks/Assignment4.ipynb

jupyter Assignment4 Last Checkpoint: 44 minutes ago (unsaved changes)

```
    lst2 = []
    lst3 = []
    recur(p1, p2, p3, p, 3, 1, 0)
    n1.append(np.sum(np.array(lst1)))
    n2.append(np.average(np.array(lst2)))
    n3.append(np.sum(np.array(lst3)))
print("Expectation=" ,"\n", n1, "\n\n", "Ensemble_avg =", "\n", n2, "\n\n")
print("Q_list:", "\n",  q)
print()
t_avg = np.average(np.array(q))
print("Time avg =", t_avg)
```

```
Expectation=
 [3.0, 3.0, 3.0, 3.0019448100000004, 3.0068651793, 3.015020740035, 3.02627108859186, 3.0403106446961763, 3.0567884185399334, 3.075362802247204, 3.0957236285560183, 3.117598657264385, 3.1407530620455897, 3.1649860729094264, 3.1901267354682807]

 Ensemble_avg =
 [3.0, 3.0, 3.0, 3.0123456790123457, 3.0329218106995883, 3.0603566529492454, 3.0928212162780064, 3.129096174363664, 3.168216227201138, 3.2094701011024744, 3.252315873257803, 3.296341456530452, 3.3412294748303824, 3.3867338466964765, 3.432662222983256]


Q_list:
 [2, 2, 3, 4, 4, 4, 4, 5, 4, 3, 2, 2, 1, 2, 3, 3, 3, 4, 5, 4, 4, 4, 3, 3, 3, 3, 2, 3, 2, 3, 3, 2, 3, 2, 2, 1, 1, 0, 0, 0, 0, 1, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7, 6, 6, 6, 5, 5, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 5, 4, 3, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4, 5, 5, 4, 4]

Time avg = 3.297029702970297
```

In [ ]:
```
def new(a):
    global a
    a = a + 1
```

In [ ]:
```
a = 5
new(a)
a
```

```
    recur(p1, p2, p3, p, 3, 1, 0)
    n1.append(np.sum(np.array(lst1)))
    n2.append(np.average(np.array(lst2)))
    n3.append(np.sum(np.array(lst3)))
print("Expectation=" ,"\n", n1, "\n\n", "Ensemble_avg =", "\n", n2, "\n\n")
print("Q_list:", "\n",  q)
print()
t_avg = np.average(np.array(q))
print("Time avg =", t_avg)
```

```
a

Expectation=
 [2.8, 2.6000000000000005, 2.4000000000000004, 2.2053144100000006, 2.0207793431000005, 1.8494532422550007, 1.6927516430837806, 1.5509115383522123, 1.4234290924781883, 1.3093860218083166, 1.2076685089403187, 1.1171043187417729, 1.036543584563181, 0.9649029368565314, 0.9011867120242532]

 Ensemble_avg =
 [3.0, 3.0, 3.0, 3.0123456790123457, 3.0329218106995883, 3.0603566529492454, 3.0928212162780064, 3.129096174363664, 3.168216227201138, 3.2094701011024744, 3.252315873257803, 3.296341456530452, 3.3412294748303824, 3.3867338466964765, 3.432662222983256]


Q_list:
 [2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 2, 1, 1, 2, 2, 3, 3, 3, 3, 2, 3, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Time avg = 0.6633663366336634
```

In [71]:

```
print ('\033[1m' + 'c'+'\033[0m')
print()
a = .9
b = .7
q = [2]
p = 100
```

Hor | Assi | Lect | Lect | Cou | Inve | how | pyti | Ran | Ens | LEC | tim | (11) | prir | Hov

localhost:8888/notebooks/Assignment4.ipynb

Apps | Microsoft Store - G... | Computer Centre | Dashboard | external asc | (2) WebMail IITB :: L... | Welcome to ASC ! | IIT Bombay Services | DownGit | BodhiTree | Steam inhalation or... | Reading list

jupyter Assignment4 Last Checkpoint: 42 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run | Code

```
print("Q_list:", "\n", q)
print()
t_avg = np.average(np.array(q))
print("Time avg =", t_avg)
```

```
b

Expectation=
 [3.2, 3.4000000000000004, 3.6000000000000014, 3.800024010000002, 4.000089077100002, 4.200201143775002, 4.4003580020173825, 4.600552954361776, 4.800777537528715, 5.001023190273921, 5.201282131208144, 5.401547731395789, 5.601814595305203, 5.802078489657688, 6.002336204201138]

 Ensemble_avg =
 [3.0, 3.0, 3.0, 3.0123456790123457, 3.0329218106995883, 3.0603566529492454, 3.0928212162780064, 3.129096174363664, 3.168216227201138, 3.2094701011024744, 3.252315873257803, 3.296341456530452, 3.3412294748303824, 3.3867338466964765, 3.432662222983256]


Q_list:
 [2, 3, 4, 5, 5, 6, 6, 7, 7, 8, 8, 8, 9, 9, 9, 10, 10, 10, 11, 12, 13, 13, 13, 14, 14, 14, 14, 15, 15, 16, 16, 16, 15, 15, 16, 16, 17, 17, 17, 17, 17, 17, 17, 16, 15, 16, 16, 17, 18, 19, 20, 20, 20, 21, 20, 20, 20, 21, 21, 22, 22, 22, 22, 23, 23, 22, 22, 22, 23, 23, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 25, 25, 25, 26, 26, 26, 27]

Time avg = 17.871287128712872
```

In [69]:

```
print ('\033[1m' + 'c'+'\033[0m')
print()
b = .7
q = [2]
p = 100
p_list = list(range(1, 16))
n1 = []
n2 = []
n3 = []
E = [2]
t_avg, ensemble_avg = 0, 0
```

ENG IN 23:32 02-11-2021

classmate

Date ______
Page ______

Question 4:

X_n is a random sequence taking values in $\{1, 2, 3, 4, 5, 6\}$. with probability

$$P(X_n = j \mid X_{n-1} = i) = p_{ij}$$

To check stationarity & ergodicity, we check the time average and ensemble average.

(a) Here, the time average is constant. Thus the process is stationary.
The process is not ergodic, because in some cases time average $\neq$ ensemble average.

(b) Here, mean is constant irrespective of time $\Rightarrow$ stationary process.
time avg $\neq$ ensemble avg $\Rightarrow$ not ergodic

(c) The ensemble mean is not constant (different for different cases).
$\Rightarrow$ Not stationary $\Rightarrow$ not ergodic

(d) Mean takes a constant value, irrespective of time.
$\Rightarrow$ stationary process
However, ensemble avg $\neq$ time avg
$\Rightarrow$ not ergodic.