Kemal Yagız Daşkıran
040210024

1) a) If bit error occur in 2 successive bits, than there can be 4 different scenerios.

1) $b_3 \to b_2$ error : $2^3 \cdot \Delta + 2^2 \cdot \Delta = 12\Delta$

2) $b_2 \to b_1$ error : $2^2\Delta + 2\Delta = 6\Delta$

3) $b_1 \to b_0$ error : $2\Delta + \Delta = 3\Delta$

4) $b_0 \to b_3$ error : $\Delta + 2^3\Delta = 9\Delta$ (like 1001 1001 → 1000 0001 resulting in a $b_0 b_3$ error)

$$E[n_d^2] = \frac{1}{4}(144\Delta^2 + 36\Delta^2 + 9\Delta^2 + 81\Delta^2) = \underline{67{,}5\Delta^2}$$

b) Maximum Error in volts is the $b_3 b_2$ error resulting in a $\underline{12\Delta V}$ error.

c) Gray Coding can be used to reduce the maximum error on a sample. Dropping the value to 8Δ from 12Δ.

Decimal	Gray Coding
15	1000
14	1001
13	1011
...	
7	0100

even if $b_3 b_2$ MSB error occurs, only $8\Delta V$ error would occur. Therefore decreasing the maximum error.

d) $b_3 b_2$ error : 8Δ, $b_2 b_1$ error : 6Δ, $b_1 b_0$ error : 3Δ

$$\frac{64\Delta^2 + 36\Delta^2 + 9\Delta^2}{3} = \frac{109\Delta}{3} = \underline{36{,}3\Delta^2}$$

Noise has decreased with the usage of Gray Coding

Kemal Yağız Daşkıran
040210034

2) $p_{RC}(t) = \text{sinc}\left(\frac{t}{T}\right) \cdot \dfrac{\cos\left(\pi a \frac{t}{T}\right)}{1 - 4a^2\left(\frac{t}{T}\right)^2}$

1)

$\text{sinc}(x) = \dfrac{\sin(\pi x)}{x}$ ise $\text{sinc}(x - \frac{1}{2}) = \dfrac{\sin(\pi(x - \frac{1}{2}))}{\pi(x - \frac{1}{2})} = \dfrac{\sin(\pi x - \frac{\pi}{2})}{\pi(x - \frac{1}{2})}$

if:

$\sin(\pi x - \frac{\pi}{2}) = -\cos(\pi x)$ then $\rightarrow$ $\text{sinc}(x - \frac{1}{2}) = \dfrac{-\cos(\pi x)}{\pi(x - \frac{1}{2})}$

2) $\text{sinc}(x + \frac{1}{2}) \rightarrow$ Same as $\text{sinc}(x - \frac{1}{2}) \rightarrow \text{sinc}(x + \frac{1}{2}) = \dfrac{\cos(\pi x)}{\pi(x + \frac{1}{2})}$

$$\text{sinc}(x + \tfrac{1}{2}) + \text{sinc}(x - \tfrac{1}{2}) = \frac{\cos \pi x}{\pi}\left(\frac{-1}{x - \frac{1}{2}} + \frac{1}{x + \frac{1}{2}}\right)$$

$$= \left(\frac{-x - \frac{1}{2} + x - \frac{1}{2}}{(x - \frac{1}{2})(x + \frac{1}{2})}\right) \cdot \frac{\cos \pi x}{\pi} = \frac{\cos \pi x}{\pi}\left(\frac{-1}{x^2 - \frac{1}{4}}\right) = \text{sinc}(x - \tfrac{1}{2}) + \text{sinc}(x + \tfrac{1}{2})$$

$$= \frac{\cos \pi x}{\pi(\frac{1}{4} - x^2)} \longrightarrow \frac{\pi}{4}\left(\text{sinc}(x - \tfrac{1}{2}) + \text{sinc}(x + \tfrac{1}{2})\right) = \frac{\cos(\pi x)}{1 - 4x^2}$$

if we say $x = \alpha \frac{t}{T}$ then $\boxed{\dfrac{\cos \pi \cdot \alpha \frac{t}{T}}{1 - 4\alpha^2 \frac{t^2}{T^2}} = \dfrac{\pi}{4}\left(\text{sinc}(\alpha \tfrac{t}{T} - \tfrac{1}{2}) + \text{sinc}(\alpha \tfrac{t}{T} + \tfrac{1}{2})\right)}$

if we use this on our initial formula, then:

$$\boxed{p_{RC}(t) = \frac{\pi}{4} \cdot \text{sinc}\left(\frac{t}{T}\right) \cdot \left[\text{sinc}\left(\alpha \frac{t}{T} - \frac{1}{2}\right) + \text{sinc}\left(\alpha \frac{t}{T} + \frac{1}{2}\right)\right]}$$

3. Soru

My MATLAB Code for the third question has shown below:

3.a)

```
N = 10000; % Number of bits

bits = randi([0,1], N,1);
```

3.b)

```
a_k = 2*bits - 1; % Mapping 0-1's to -1,1
```

3.c)

```
L = 4;

X_upsampled = upsample(a_k,L);

M = X_upsampled;
```

3.d)

```
gT = [1 1 1 1];

Xc = conv(X_upsampled,gT,'Same'); % Generating the Xc
```

3.e)

```
nw = randn(40000,1); % Generating White Noise

y = Xc + nw;
```

3.f)

```
sampled_y = downsample(y,L);
```

3.g)

```
detected_bits = sampled_y > 0;   %  Created the Decision rule

DecisionRule_n_of_errors = sum(detected_bits ~= bits);

BER = DecisionRule_n_of_errors/length(bits);
```

```
BER =

    0.1571
```

3.h)

```
Matched_Filter = [1 1 1 1];

r = conv(y, Matched_Filter, 'Same');

mf_sampled_y = downsample(r,L);

mf_detected_bits = mf_sampled_y > 0; %  Created the MF Decision rule

MF_n_of_errors = sum(mf_detected_bits ~= bits) % Number of errors if we use a MF

MF_BER = MF_n_of_errors/length(bits)
```

```
MF_n_of_errors =

   907


MF_BER =

    0.0907
```

% BER found with a MF is reasonabily better than without the usage of a

% matched filter.

3.i)

%V1 = 4 V0 =-4 Therefore d = 8

%N = 2

BER_I = qfunc(2)

```
BER_I =

    0.0228
```

3.j)

%When we use MF on the Noise, N becomes 4

BER_J = qfunc(1)

```
BER_J =

    0.1587
```

Name ▲	Value
a	0.1587
a_k	*10000x1 double*
BER	0.1632
BER_I	0.0228
BER_J	0.1587
BER_MF	0.1587
BER_without	0.0228
bits	*10000x1 double*
DecisionRule_...	1632
detected_bits	*10000x1 logical*
gT	[1,1,1,1]
L	4
M	*40000x1 double*
Matched_Filter	[1,1,1,1]
MF_BER	0.0904
mf_detected_b...	*10000x1 logical*
MF_n_of_errors	904
mf_sampled_y	*10000x1 double*
N	10000
nw	*40000x1 double*
r	*40000x1 double*
sampled_y	*10000x1 double*
V1	[-1,-1,-1,-1]
X_upsampled	*40000x1 double*
Xc	*40000x1 double*
y	*40000x1 double*

Code, without any interruption looks like this:

N = 10000; % Number of bits

bits = randi([0,1], N,1);

a_k = 2*bits - 1; % Mapping 0-1's to -1,1

L = 4;

X_upsampled = upsample(a_k,L);

Kemal Yağız Daşkıran 040210034

```
M = X_upsampled;


gT = [1 1 1 1];

Xc = conv(X_upsampled,gT,'Same'); % Generating the Xc


nw = randn(40000,1); % Generating White Noise


y = Xc + nw;


sampled_y = downsample(y,L);


detected_bits = sampled_y > 0; %  Created the Decision rule

DecisionRule_n_of_errors = sum(detected_bits ~= bits);

BER = DecisionRule_n_of_errors/length(bits);


Matched_Filter = [1 1 1 1];

r = conv(y, Matched_Filter, 'Same');


mf_sampled_y = downsample(r,L);


mf_detected_bits = mf_sampled_y > 0; %  Created the MF Decision rule

MF_n_of_errors = sum(mf_detected_bits ~= bits) % Number of errors if we use a MF

MF_BER = MF_n_of_errors/length(bits)


% BER found with a MF is reasonabily better than without the usage of a

% matched filter.


%V1 = 4 V0 =-4 Therefore d = 8

%N = 2

BER_I = qfunc(2)


%When we use MF on the Noise, N becomes 4

BER_J = qfunc(1)
```