(1)

Computation of $\frac{E_b}{N_0}$:

Our modulation is BPSK

i.e., signal power = 1

Rate of turbo code is 1/3

i.e., for one bit transmission we use 3 BPSK signals

- BPSK modulated signals have power 1

3 signals have power 3 → R is code Rate

In general $E_s = R E_b \rightarrow E_b = 3E_s$

$\downarrow 1/3$ $\downarrow 1$

$E_b = 3$ in our case

$$\frac{E_b}{N_0}\,dB = 10\log_{10}\frac{E_b}{N_0} \rightarrow \frac{E_b}{N_0} = 10^{\frac{E_b}{N_0}\,dB \times \frac{1}{10}}$$

$$\sigma^2 = \frac{N_0}{2} \rightarrow \frac{E_b}{N_0} = 10^{\frac{E_b}{N_0}\,dB \times \frac{1}{10}}$$

$E_b = \frac{1}{R}E_s$

$$\frac{\frac{1}{R}E_s}{2\sigma^2} = \underbrace{10^{\frac{E_b}{N_0}\,dB \times \frac{1}{10}}}_{E_b/N_0}$$

$E_s \rightarrow 1$ for BPSK

$$\Rightarrow \sigma^2 = \frac{}{2R\frac{E_b}{N_0}}$$

$$\boxed{\sigma^2 = \frac{1}{2R\,E_b/N_0}}$$

(1) Turbo Code Simulations

$\frac{\overline{Eb}}{No}$ dB = [0 0.1 0.2 ... 4]

for(indx=0; indx < length($\frac{\overline{Eb}}{No}$ dB); indx++)
{

$\frac{Eb}{No}$ dB = $\frac{\overline{Eb}}{No}$ dB(indx), ← get a value of $\frac{\overline{Eb}}{No}$ dB

$\frac{Eb}{No} = 10^{\frac{\frac{Eb}{No} \times \frac{1}{10}}{}}$; ← calculate real $\frac{Eb}{No}$

codeRate = 1/3 ← Turbo code rate without trellis termination

$\sigma^2 = \frac{1}{2 \cdot codeRate \times \frac{Eb}{No}}$ → noise variance (noise power)

initialize variables here ←

do{ → enter to a loop

data ← generate data bit frame

codeword ← turbo encode data

signal ← BPSK modulate codeword

noise ← generate noise with variance σ^2 (noise has zero mean, Gaussian)

receivedsignal = signal + noise

resolved ← decode receivedsignal and decide on the transmitted bits

difference = data - resolved;

sm = sum(abs(difference));

(2)

```
If sm != 0  <- if sm not equal to zero
{
 tot_bit_error = tot_bit_error + sm  <- add it to total bit error
 errFrame++;  <- increment error frame number
   frmNumber++  <- increment data frame number
 }
 else
  {
    frmNumber++
  }
} while (errFrame<50)
ber = tot_bit_error / frmNumber * 1024
  lgber = log ber;
            10
  BER(indx) = lgber;
```

} → end of the for loop

initialize variables include the following

tot_bit_error=0; frmNumber=0

errFrame=0;

(3) Trellis termination for Turbo Encoders: