HW5 Convolutional code

Dec. 5, 2023

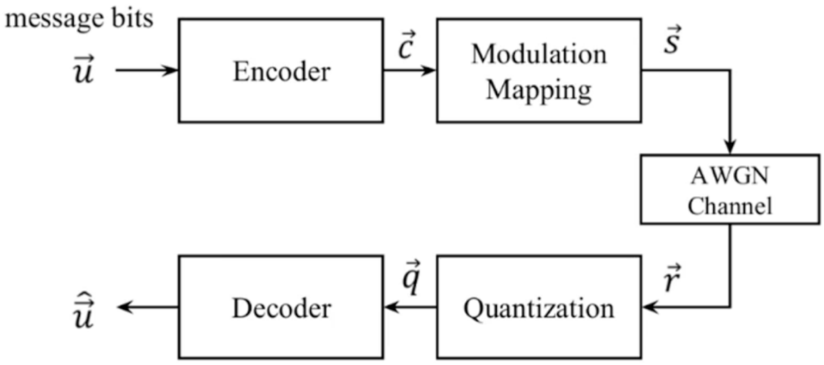
Due date Dec. 22, 2023

Consider a (2,1,4) convolutional code with generator sequences $(2,3)_8$ and $(3,5)_8$ in octal form respectively. Use the Viterbi algorithm to obtain its BER performance over the AWGN.

(a) The truncation length is set to $\tau = 32$ blocks. Three BER curves with $Q = 2,\ 4,\ 8$ respectively are plotted against the E_b/N_0 ranging from 2 dB to 10 dB for every increment of 0.5 dB.

(b) The truncation length is set to $\tau = 12$ blocks. Three BER curves with $Q = 2,\ 4,\ 8$ respectively are plotted against the E_b/N_0 ranging from 2 dB to 10 dB for every increment of 0.5 dB. .

System model



Code Architecture

```
for j = progress(1:1:3)
    Q = 2^j;
    for snr = 2:1:10
        err=0;
        for i = 1:10000
            frame = frame+1;
            u = randi(2, 1, len) - 1;
            c = Encoder(u, len);
            s = Modulation(c, len, Eb);
            r = awgn(s, snr);
            q = Quantization(r, Q, len, Eb);
            d = Demodulation(r, len);
            trellis = poly2trellis(5, [23,35]);
            y = Decoder(q, trellis, len);
            err = err + ErrorCalculate(u, y, len);
        end
        ber(j, snr-1) = err/(frame*len);
    end
end
```

Simulation

- Transmitter
  - Encoder: (n, k, m) convolutional code.
  - Modulation: BPSK: 0 --> sqrt(Eb), 1 --> -sqrt(Eb)
- AWGN: r = s + n
  - Noise sample n are i.i.d. Gaussian distribution with zero mean and variance $N_0/2$.
  - Implementation: https://www.mathworks.com/help/comm/ref/awgn.html
- Decoder
  - Using the Viterbi algorithm with soft decisions to decode the received signals.

- Save the result in a text file named "result(a).txt" and "result(b).txt".

The format is like:

```
(2, 1, 4)convolutional code with Q = 2/4/8 and truncation length = 32
SNR = 2(dB), BER = 0.0510758/ 0.0188995/ 0.0110992
SNR = 2.5(dB), BER = 0.0350054/ 0.0110486/ 0.0055087
SNR = 3(dB), BER = 0.0226734/ 0.00602645/ 0.00249145
SNR = 3.5(dB), BER = 0.0134534/ 0.0030947/ 0.00100995
SNR = 4(dB), BER = 0.00761465/ 0.00148905/ 0.0003859
SNR = 4.5(dB), BER = 0.0039159/ 0.0006867/ 0.00012705
SNR = 5(dB), BER = 0.0018793/ 0.0003092/ 3.92e-05
SNR = 5.5(dB), BER = 0.00083665/ 0.0001192/ 9.85e-06
SNR = 6(dB), BER = 0.000339/ 5.075e-05/ 3.05e-06
SNR = 6.5(dB), BER = 0.0001277/ 1.735e-05/ 4e-07
SNR = 7(dB), BER = 4.295e-05/ 5e-06/ 2e-07
SNR = 7.5(dB), BER = 1.745e-05/ 1.7e-06/ 0
SNR = 8(dB), BER = 4e-06/ 3e-07/ 0
SNR = 8.5(dB), BER = 7.5e-07/ 1e-07/ 0
SNR = 9(dB), BER = 3.5e-07/ 0/ 0
SNR = 9.5(dB), BER = 0/ 1e-07/ 0
SNR = 10(dB), BER = 0/ 0/ 0
```

- Using the function "semilogy" to plot the BER performance and save it as "BER(a).jpg" and "BER(b).jpg.
- Remember to add the labels for the x-axis and y-axis and the legends of each curve.
- All your coding files (.m 檔), text files, and .jpg files 壓縮至一個壓縮檔 "r11942139_hw5.zip"並上傳至 ntu cool.