ECE 6390 Solutions to Homework 6: Digital Modulation and Spread Spectrum

1. Error-Correction Coding:

The average bit error rate (BER) of BPSK modulation at C/N = 7 dB from the BER curves in textbook is $\approx 10^{-3}$.
With Turbo coding, the coding gain would be the difference in dB that yields the same BER of 10^{-3} as the uncoded BPSK. From BER curve, the value of Turbo that gives BER of 10^{-3} is approximately 0.8 dB. Therefore, $coding\_gain = (7 - 0.8)dB = 6.2dB$ for using Turbo codes!
To find this distance, assume that the other parameters (frequency, ES receiver noise, etc) remain the same as before. Using the path-loss equation, the carrier power:

$$\begin{aligned} C &= P_R = \frac{P_T G_T G_R}{\left(\frac{4\pi r}{\lambda}\right)^2} \\ &= \frac{P_T G_T G_R}{\left(\frac{4\pi}{\lambda}\right)^2 \dot{r}^2} \text{ for geostationary, } r = 35,786 \times 10^3 m \end{aligned} \quad (1)$$

We define a constant, $\Gamma = \frac{P_T G_T G_R}{\left(\frac{4\pi}{\lambda}\right)^2}$. Then, we know that at 7 dB, the following held for the BPSK:

$$\begin{aligned} 10^{7/10} &= \frac{\Gamma}{(35,786 \times 10^3)^2} \\ \Rightarrow \Gamma &= 10^{7/10} \times (35,786 \times 10^3)^2 \end{aligned} \quad (2)$$

At 0.8 dB (the value that gives same BER as uncoded BPSK), we have:

$$\begin{aligned} r &= \sqrt{\frac{\Gamma}{10^{0.8/10}}} = \sqrt{10^{6.2/10} \times (35,786 \times 10^3)^2} \\ &= \approx 73,066\ km \end{aligned}$$

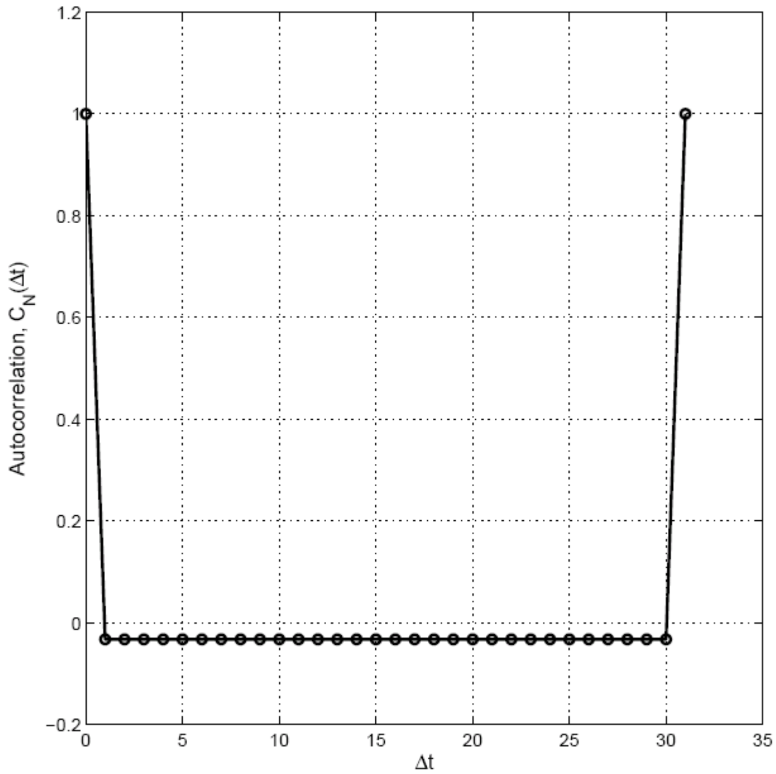
2.

(a) The MATLAB code used to generate the ouput is shown below:

```
% This problem solves part 1 of Homework #7
clear, close all

register = ones(50,5); % initializes ones in all register contents
register(1,:) = zeros(1,5); % initial content of registers at first step = 0

% compute next state
register(2,1) = not(register(1,5));
register(2,2) = register(1,1);
register(2,3) = register(1,2);
register(2,4) = xor(register(1,3), not(register(1,5)));
register(2,5) = register(1,4);

index = 2;
while (index < size(register,1)) & (sum(not(xor(register(index,:), register(1,:)))) ~= 5)
    index = index + 1;
    register(index,1) = not(register(index-1,5));
    register(index,2) = register(index-1,1);
    register(index,3) = register(index-1,2);
    register(index,4) = xor(register(index-1,3), not(register(index-1,5)));
    register(index,5) = register(index-1,4);
end

if index == size(register,1)
    display('Increase the number of simulation runs')
else
    Tcode = index - 1;
    Output = register(1:index-1,end);
    register = register(1:index-1,:);
    xt = 2*Output-1; % converts the binary 1 and 0 into antipodal values of 1 and -1 resp.
    xt = xt(:)'; % convert into row vector
    display(['The code''s period is: ', num2str(Tcode)])

    % computer autocorrelation values
    R = zeros(1,Tcode);
    for i = 0:Tcode
        R(i+1) = (1/Tcode)*(xt * [xt(end-i+1:end), xt(1:end-i)]');
        %R(i+1) = (1/Tcode)*(xt * [zeros(1,i), xt(1:end-i)]');
    end
    figure, plot([0:Tcode],R), xlabel('\Deltat'), ylabel('Autocorrelation, C_N(\Deltat)')
    grid on
end
```

With this code, one period of the output sample is: 0 0 1 1 0 1 0 0 1 0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 and the code period, $T_{code} = 31$ bits.

(b) The autocorrelation function produced is shown below:



As can be seen from this figure, the code is not ideal because the autocorrelation function is not ideally zero when the shift, $\Delta t \neq 0$. In other words, the autocorrelation does not follow the Dirac delta function.