ENGIN 112: Module 3 Homework

Due: 5:00pm, Friday, October 7, 2022

Question 1

(This question refers to the relationship between the frequency of a transmitted signal and the antenna length that is needed.)

Find the best transmitting antenna lengths for the following communication systems:

(a) An FM radio station with a carrier frequency of 105.5 MHz .

(b) A Wi-Fi transmitter using a carrier frequency of 2.4 GHz .

(c) An indoor communication system with a carrier frequency of 60 GHz .

Question 2

(This question asks you to find how much of the power transmitted by a radio station's antenna is picked up by a radio receiver's antenna located some distance away. *(Hint: it's not very much!)*)

If a transmitting antenna radiates uniformly in all directions, then (assuming that we use a frequency with negligible atmospheric attenuation - see Question 3) at a distance of r meters from the antenna the transmitted power is distributed evenly over a sphere of radius r centered at the antenna. (Recall that the surface area of a sphere with radius r meters is $4\pi r^2$ m^2.) Say that an AM radio station is broadcasting with a transmitted power of 10 kilowatts. We have a radio receiver at a distance of 20 kilometers from the station. The receiver's antenna picks up the transmitted power in the portion of the propagating wave that hits its antenna. If the total area of the receiver's antenna facing the propagating wave is 50 cm^2, and if the radio station's antenna transmits uniformly in all directions, how much power do we receive from the radio station?

Question 3

(This question asks you to consider how frequency-dependent power loss called *attenuation* affects transmission of EM waves.)

We've seen how the choice of carrier frequency in a communication system affects transmitting antenna length. Several other important communication system characteristics also depend on the carrier frequency. One important example is **attenuation** of an EM wave as it propagates through the atmosphere. Attenuation is power loss due to the absorption of electromagnetic energy by components, especially oxygen and water, in the atmosphere. The rate of attenuation depends on the carrier frequency. A plot of atmospheric attenuation vs. carrier frequency in the GHz range is shown on the next page. (Note that both axes for the plot are on logarithmic scales.) The solid green curve shows attenuation due to oxygen and the dotted red curve shows attenuation due to water. You can assume that any other cause of attenuation is negligible, and that the total attenuation is the sum of that due to oxygen and that due to water.

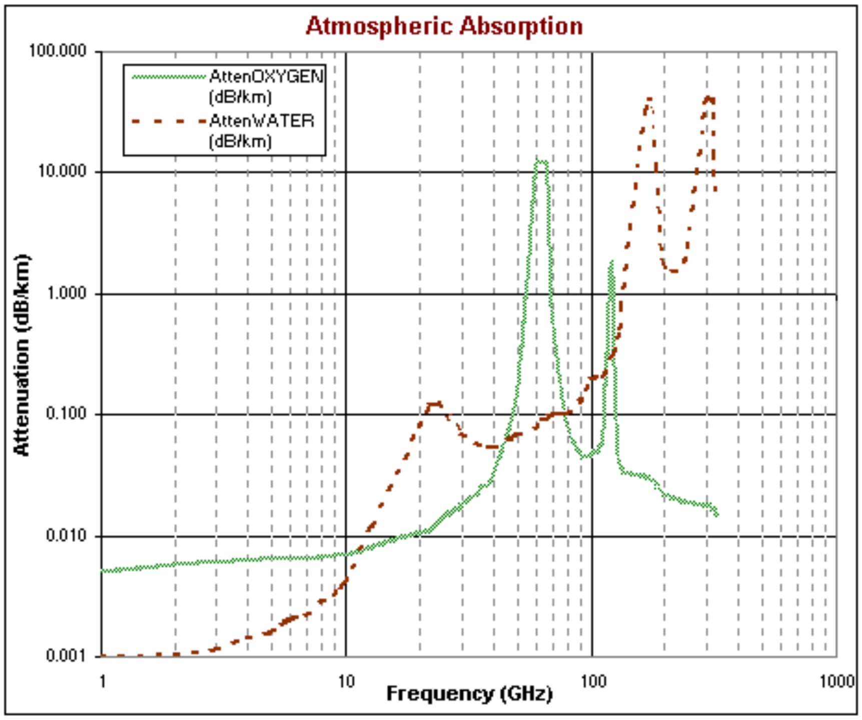

Figure 1: Atmospheric attenuation as a function of carrier frequency.

The attenuation level is given on the y-axis in units of dB/km - for example, if the total attenuation is 1 dB/km, that implies that at a point 5 km from the transmitter there will be a 5 dB loss in power due to attenuation. Let P_1 be the transmitted power, and P_2 the power some distance away from the transmitter. Note that if $10\log_{10}(P_2/P_1) = x$ dB, then $P_2 = (10^{x/10}) \cdot P_1$. So a power loss of 5 dB (that is, $x = -5$) means that power is down to $10^{-5/10} = 0.316$ of the transmitted power - that is, almost 70% of the transmitted power has been lost to attenuation.

(a) Three frequencies that have been approved by the FCC for Wi-Fi systems are 2.4 GHz, 5 GHz, and 60 GHz. For each of those frequencies, find the percentage of transmitted power that is lost to attenuation when we are 1 km away from the transmitter.

(b) Use your answer to part (a) to give one reason why 60 GHz systems are intended primarily for use inside single buildings (e.g., warehouses).

Question 4

(This question asks you to consider how multiplication of a signal by a sinusoid accomplishes frequency shifting, and how that is used in actual AM receiver design.)

Suppose that we have a message signal $m(t)$ having the spectrum shown on the next page:

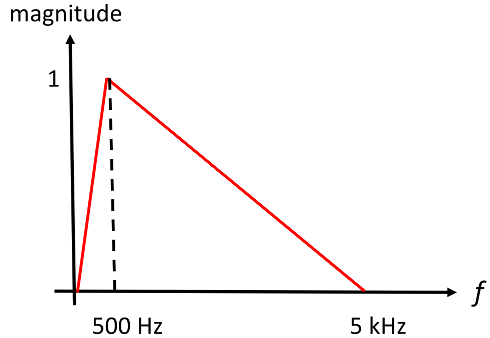

Figure 2: Spectrum of message signal.

(a) We want to transmit the message using the AM signal

$$x_{AM}(t) = [A_x + m(t)] \cdot \cos(2\pi f_c t) \tag{1}$$

where the carrier frequency is $f_c = 1$ MHz. Draw a picture of the spectrum of $x_{AM}(t)$ What are the maximum and minimum frequencies in $x_{AM}(t)$?

(b) Now, suppose the tunable BPF of an AM receiver is tuned to receive this x_{AM}. What is the frequency range of this tuned BPF?

Question 5

(This question looks at how the bit rate in a communication system is related to transmission bandwidth and the signal-to-noise ratio.)

As noted in class, one big concern in a digital communication systems is the **bit rate** R bits/sec that the system can transmit. Bits are transmitted using electrical pulses. Pulses that have shorter time duration occupy larger frequency bands. So, we can use shorter pulses and send more bits/sec if the **transmission bandwidth** B (Hz) allowed for the communication system is large. On the other hand, noise in the system makes it more difficult to transmit bits accurately. If the signal-to-noise ratio (SNR) is small, then the pulse for each bit has to be long so that enough signal energy can build up at the receiver to overcome the noise - this has the effect of reducing the bit rate. The **Shannon-Hartley Formula** for the maximum bit rate that can be achieved for a given bandwidth and SNR is

$$R \leq B \cdot \log_2(1 + SNR) \tag{2}$$

(*Note:* In this formula, SNR is **not** written in dB. Also, $\log_2(x) = \log_{10}(x)/\log_{10}(2)$.)

(a) A standard Wi-Fi channel has a bandwidth of $B = 40$ MHz. If we use that channel, what SNR (in dB) do we need in order to achieve a maximum bit rate of 100 Mb/sec? What SNR (in dB) would allow a maximum bit rate of 400 Mb/sec?

(b) The Mars Orbiter has a transmission bandwidth of $B = 100$ kHz. The maximum SNR (in dB) for a transmission from the orbiter to earth is 8 dB. What is the maximum possible bit rate for communication from the orbiter to earth?

Question 6

(In this problem we use MATLAB to simulate a receiver for a digital communication system and to see how the system's performance is affected by the noise level.)

We want to use a BPSK digital communication system to transmit bits over a wireless channel. Suppose that we want to send $R_B = 1 \times 10^5$ bits/sec (that is, 100 kb/sec). To do this using BPSK, we define a carrier signal having a duration $T_B = \frac{1}{R_B} = 1 \times 10^{-5}$ sec, and a frequency $f_c = 10R_B = 1$ MHz. (In general, the frequency needs to be an integer multiple of T_B^{-1} - in this problem we set the multiple to be 10.) In this problem we set the power of the carrier signal to be 1 watt and so set the amplitude of the carrier to be $\sqrt{2}$ volts. So, our carrier is

$$c(t) = \sqrt{2}\cos(2\pi(1 \times 10^6)t). \tag{3}$$

To send bit value “1” in a given bit interval, we transmit $c(t)$ in the interval; to send bit value “0”, we transmit $-c(t)$. Let $x(t)$ denote the transmitted signal. Random noise is added to $x(t)$ during transmission, so the received signal is $r(t) = x(t) + n(t)$. In this problem we will use a MATLAB simulation to see how the noise power level affects the receiver’s error rate in recovering the transmitted bits.

(a) First we create the carrier $c(t)$. Again, since MATLAB is a program, we have to use sampled signals. Define the bit interval by

```
TB = [0:1e-7:(1e-5)-(1e-7)];
```

(This creates an interval of length 1×10^{-5} sec (denoted 1e-5 in MATLAB) with 100 sample times in the interval.) Then we define the carrier by

```
c=sqrt(2)*cos(2*pi*(1e6)*TB);
```

Say that we want to create the transmitted signal $x(t)$ for the 5-bit sequence 10110; then we form

```
x=[c, -1*c, c, c, -1*c];
```

To create a random noise signal $n(t)$ having the same duration as $x(t)$ (that is, 5 bit intervals) we use the commands

```
T=[0:1e-7:(5e-5)-(1e-7)];
n=randn(size(T));
```

(The command **randn(...)** generates a set of *normal* (also called *Gaussian*) random numbers.) The noise signal created by this command has a normalized power equal to 2/(number of samples in a bit period) - in this case, $2/100 = 0.02$ Watt. So, to have a power of P watts in the noise signal, we multiply the noise by the factor $\sqrt{50P}$. For example, suppose that we want to simulate a received signal for which the signal-to-noise ratio (SNR) = 5. Then since the transmitted signal power was set at 1 watt, we need the noise power to be 0.2 watt. The received signal with that noise power level can then created as

```
r1=x+sqrt(50*0.2)*n.
```

Similarly, we let r0 represent the received signal when there is no noise (that is, r0 = x), and r2 represent the received signal when SNR = 1 (so, the noise power level is 1 watts, and r2 = x + sqrt(50)*n). Plot the received signal with: (i) no noise; (ii) SNR = 5; and (iii) SNR = 1, using the commands:

```
subplot(3,1,1),plot(T,r0,'k')
subplot(3,1,2),plot(T,r1,'b')
subplot(3,1,3),plot(T,r2,'r')
```

(Use the **axis** command as described in Module 2 homework to get good scales for the plots.) Attach a copy of your plot to your homework. How does the noise power level affect your ability to distinguish the transmitted signals for bits "1" and "0"?

(b) Now we will simulate a receiver acting on the received signals. As described in class, the optimal receiver for recovering the transmitted bit value in a given bit interval first multiplies the received signal in that interval by $c(t)$ (that is, we form $y(t) = r(t)c(t)$). The receiver then averages (integrates) $y(t)$ over the bit interval. If the result is a number ≥ 0, the receiver decides that a "1" was sent; if the result is a number < 0, the receiver decides that a "0" was sent. Start with the noiseless received signal, and use the following commands:

```
r=r0;
br=[];
for k=1:5
rk=r(100*(k-1)+1:100*k);
yk=sum(rk.*c);
brk=(yk>=0);
br=[br,brk];
end
br
```

This set of commands first sets the received signal to be r0, and sets up an empty vector br for the bits recovered by the receiver. In the "for loop", we set rk to be the received signal in the k^{th} bit interval ($k = 1, \ldots, 5$), and yk to be the summed (that is, integrated or averaged) value of the product of rk and c. The line defining brk returns the value 1 if the value of yk is ≥ 0, and the value 0 otherwise. This is the receiver's estimate of the k^{th} transmitted bit. The last line in the for loop fills up the vector br with the receiver's bit estimates. After the end of the for loop we print the values in br. If the receiver makes no mistakes, we should have br = 10110 (the transmitted bits). Run this for r=r0, r=r1, and r=r2, and write the recovered bit pattern in each case. Did the receiver make any errors?

(c) Finally, we will simulate sending a large number of bits (10 kbits) so we can get a good estimate of the error rate we would expect with different values of SNR. We first generate a random set of $10,000$ bits to be transmitted using the command

```
b=randi(2,1,10000)-1;
```

(Try looking at a range of samples from b, for example b(2001:2015), to verify that you are generating bits in a random-looking pattern.) Next, similarly to part (b), we will use a for loop to simulate transmitting each bit, adding noise, and recovering bits at the receiver. We will store the number of incorrectly recovered bits in a variable nerror. To initialize the simulation loop, we specify the noise power P and set nerror = 0. For example, if we want to consider the noiseless case (noise power = 0), we use the commands

```
P=0;
nerror=0;
for k=1:10000
xk=(2*b(k)-1)*c;
nk=randn(size(xk));
rk=xk+sqrt(50*P)*nk;
yk=sum(rk.*c);
brk=(yk>=0);
nerror=nerror+abs(brk-b(k));
end
```

```
ber=nerror/10000
```

The last line in the for loop increments nerror by adding the absolute value of the recovered k^{th} bit minus the true transmitted bit (so if the recovered bit is correct we add 0 to the value of nerror, and if the recovered bit is wrong we add 1). The **bit error rate** (ber) is the number of bit errors divided by the number of transmitted bits (in this simulation, nerror/10,000). Run this simulation for (i) noise power = 0; (ii) SNR = 5; (iii) SNR = 1 (and more values, if you want), and report the bit error rates that you get. (Remember to specify P and reset nerror = 0 before each run.) Based on your results, do you think SNR = 5 is a reasonable operating condition for successful communication? What about SNR = 1?