1 Antennas: two dipoles

We consider 2 vertical dipoles (see figure 1). The distance between them is $a$.

1. What is the difference between the distances from the dipoles to any point of the horizontal plane defined by its polar coordinates $(d, \theta)$? (There is an exact answer, but there is a much simpler good approximation that you can use in the next questions) (If you are using the approximate formula, please give under which conditions it is valid)

2. What is the received amplitude at any point of the horizontal plane if the two dipoles are fed with the same signal?

3. Plot this amplitude as a function of $\theta$, for a reasonable $d$ and various well-chosen values of $a$.

4. What value of $a$ gives the pattern shown in figure 2?

5. How do you think the antenna pattern looks in the vertical plane?

6. What would we get if we used more dipoles?
2 DSSS

This problem will explore some of the noise immunity properties of direct sequence spread spectrum. You will generate ordinary square pulse bits versus spread bits and compare how each behaves when other noise/interference signals are added. The bit rate is one bit per second and we will simulate 2 seconds. Do the following steps in a spreadsheet, or matlab, or R, or… Let $C_x$ be the $x$th column and $C_x(i)$ is the value of the $i$th row in the $x$th column.

1. $C_1$ will be time: $C_1(1) = 1/110, C_1(2) = 2/110, \ldots C_1(220) = 2$

2. $C_2$ will represent a 0 followed by a 1. Fill in 110 $-1$’s followed by 110 1’s. So we have 110 “samples” per transmitted bit.

3. $C_3$ and $C_4$ will represent the bit shape. $C_3$ will represent an ordinary square pulse: fill in all 1’s. $C_4$ will represent a spread pulse as follows. Use the 11 chip Barker spreading code: $+1 -1 +1 +1 -1 +1 +1 -1 -1 -1$. Fill in with each chip repeated 10 times (i.e. 10 $+1$, 10 $-1$, 20 $+1$, etc.). Repeat the pattern twice to represent the two bits.

4. The transmitted signal is the fifth and sixth columns. $C_5 = C_2 \times C_3$ is the square pulse. $C_6 = C_2 \times C_4$ is the spread pulse.

5. In the remaining columns we will place various types of noise sources. Above the column you will put the demodulated values for the square and spread wave. Let $C_d$ be the column of the data sent, $C_p$ be the column with the pulse shape, $C_n$ be the current column, and we use the notation $C_x(i)$ to be the value in the $i$th row of column $C_x$. The output for bit 1 is

$$\frac{1}{110} \sum_{i=1}^{110} (C_d(i) + C_n(i)) \times C_p(i)$$
and similarly for bit 2. So, you should have four rows above for each of the 4 bits.

The table of noise sources is below. $C_t = C_1$ is the time value column.

<table>
<thead>
<tr>
<th>Col = Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_7(i) = 0$</td>
<td>Noise-free case</td>
</tr>
<tr>
<td>$C_9(i) = 2 \cos(10C_t(i))$</td>
<td>High frequency narrowband noise</td>
</tr>
<tr>
<td>$C_{10}(i) = 2 \cos(1.5C_t(i))$</td>
<td>Similar frequency narrowband noise</td>
</tr>
<tr>
<td>$C_{11}(i) = 2 \cos(0.5C_t(i))$</td>
<td>Low frequency narrowband noise</td>
</tr>
<tr>
<td>$C_{12}(i) = 2\text{Normal}(0, 1)$</td>
<td>Noise</td>
</tr>
</tbody>
</table>

Point the bits that are corrupted. For the purposes of this experiment, corrupted is defined as being the wrong sign or within 0.1 of the wrong sign. Which method has the least number of errors? What is your intuitive explanation for why this is true?

In working this problem, it will help to plot the different columns to see what is going on. You only need to turn in the first few rows with the decoded bit information.

You can write a program that can simulate many random bits and compute the BER in each noise case.