Homework Help - Problem Set 5

[**The Modulo Precoder (2.28)**] The modulo precoder for a specified lattice uses that lattice's Voronoi region or "shell" to surround constellation. The precoder input subsymbol vectors may be from some code (good or bad) and usually well spaced on some N-dimensional grid (maybe offset to make the mean value zero). These subsymbol vectors can be any set of points that represent the code's subsymbol output possibilities chosen from some different lattice. These points need to remain distinct from one another for that subsymbol after the precoder with the intra-codeword distances preserved. The precoder maps these points into their distinct equivalents inside the shell, but accepting an input vector (which may be any complex/real value, especially if other user xtalk is presubtracted) and finds the point at minimum distance from the Voronoi region's center point to represent the output. Indeed, this operation can implemented by a decoder (in the transmitter) like the ones you already know well.

This first PS5 problem (2.28) provides the first lattice's generators, and so all center points in the shaping lattice Voronoi regions are integer-coefficient linear combinations of these generator vectors. By trying a few integers, the problem rapidly will find the closest such point. The encoder output is the difference - this works even if the input itself is not vectors from a second lattice and just any N-dimensional value (the code may not be good, but the precoder still produces an output).

a. Convince yourself that two acceptable generators for $4\mathbb{Z}^2$ are $\boldsymbol{g}_1 = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$ and $\boldsymbol{g}_2 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$. Now try to find the closest integer multiples of these generators' sum to the point $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$. The first one is trivially within the Voronoi region already. The others are outside this region, so map via the precoder/modulo into points inside the region.

b. It's a little more effort, but don't spend too much effort - just test a few obvious points.

c. Note the D3 lattice has points that have even sums.

[**Broadcast Channel (2.29)**]

a. The first part simply repeats an example in the text and lecture with changed numbers.

b. The first part simply repeats an example in the text and lecture with changed numbers.

c. This part needs you to pose some α values and just compute a few points to draw the region.

d. What does duality tell you?

[**Vector Broadcast Channel(2.30)**] This vector BC explores primary and secondary users.

a. This is a simple scaling program. The matlab command rank.m makes it easy.

b. You can use the simple wcnoise program here first and test to see that the worst-case noise is indeed nonsingular on this channel for this input. Look for the zeroed diagonal element to find the secondary user. Look specifically at Example 2.8.5 to follow this problem. To reorder move the primary users to upper left.

c. This part is more involved but follows the steps in the class text to eventually the various transmit and receive entities.

d. The diagram is pretty full, but follows examples. The idea in these last two parts is to cause you to detail a transmitter and receiver and this should reinforce some of the concepts.

[**Duality (2.31)**]]

a. The dual MAC (an energy-sum MAC) has the same total energy as the original BC.

b. Take transpose and reverse order.

c. This part illustrates how the best user on MAC becomes worst user on dual BC and vice versa.

d. You may want to look at the website's mac2bc program to help here. This can avoid a lot of hand calculation that would be tedious. For this simple scalar channel it readily produces the dual bc energies.

Secondary User (2.32)

a. This channel is degraded. Use the bcmax program to help find the maximum sum rate.

b. Please do eliminate finite-precision issues by ensuring the matrix is singular. You can use the optimum Rwcn's psuedoinverse (the one for max rate) and premultiplt $\tilde{H}$. The rq command with third output order argument used finds the secondary user.

c. This simply builds more insight about the different user's passage and is otherwise straightforward.