Homework Help - Problem Set 5
Solutions

[**The Modulo Precoder**] The modulo precoder for a definitely specified lattice Voronoi region uses that region "shell" for the 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.

The first lattice has Voronoi regions that tesselate space, with one lattice vector at the center of each. So, the subsymbol's equivalent point is the difference between the closest center vector of the first latttice and the precoder input. See Lecture 9

These points may have (too) large average energy, so the precoder find their equivalents inside the shell. Every distinct point has an equivalent. See Lecture 9, Slide 15. The precoder input can be viewed as a second lattice point (the closest center vector) plus noise/error to the actual code subsymbol. The precoder sends the error. As long as $2^{\tilde{b}}$ errors are distinct for any subsymbol that might occur, messages and precoder outputs remain in 1-to-1 relationship. The errors are the lowest energy representation for the codeword, varying only with the choice of lattice (sometimes called a shaping lattice). The second modulo in the receiver exploits that 1-to-1 relationship to recover the original subsymbol value.

The first homework problem 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).

[**Broadcast Channel Problems**] The first BC on homework follows the example in the text, and that was in class. It just has different numbers. It's intent is to build familiarity with the same concepts presented by doing it.

The vector BC, by contrast stretches this into first finding the primary and secondary users. The simplest way to do this is with the worst-case noise program. Compute the S_{wcn} and look for zeros. The BC does not see the secondary user set change with input autocorrelation matrix (this is not necessarily true on the energy-sum MAC). Look specifically at Example 2.8.5 to follow this problem. They track well, and this problem was to help gain BC design familiarity by doing it, but using the example as a guide.

[**Interference Channel**] The interference channel appears in L11 notes and in Section 2.9 of the text. Again here the problem in homework closely parallels the example in the notes and lecture. The hope here again is that by simply changing the numbers, but otherwise working the exercise creates an opportunity to practice the concepts in the example and gain familiarity with them.

While two users, one of them is two-dimensional in Problem 2.31, so the problem also exercises the ability to index the users correctly and then seen how the two users interact even if of different dimensionality. The channel is singular which effectively makes it look like two one-dimensional users again.

As in the corrected L11 (r3), it makes most sense to evaluate the derivatives at the vertex with largest sum.

Relay Channel The relay channel decomposes into simpler channels. For Problem 2.32, it really is just one user to keep dimensionality to a minimum. Even in this simply case, the best strategy for the first-stage BC or for the last-stage MAC may not be best overall. The analysis needs to evaluate the component channels' possibilities and then try to find the sum rate across all paths from transmitter to receiver and take the best sum (where the legs are limited by the minimum on the branches collecting the legs).