Homework Help - Problem Set 7

[MAC Design (7.1)]

a. The ν is the largest one over all users, but in this case, it is the same for all in any case. Nominally, the energy/symbol grows with $\bar{N}$ times the energy per subsymbol. However, with vector DMT, the useful portion is the fraction $\overline{N}/(\overline{N}+\nu)$. The per tone energy is thus reduced by this fraction. relative to simply equating to energy per subsymbol. The channel basically requires capturing and white-noise-normalized h in a tensor, and then executing an FFT thereupon, as in the class and text examples.

b. This is a basic Vector-DMT single-user waterfilling exercise. Matlab's blkdiag command might be helpful in stacking the H matrix tones to which then svd and waterfill_gn can be applied. It is at this point still a single-user design.

c. This part returns to multiuser. Use of mu_mac is probably best here, with 8 uses in a loop to compute the GDFE receiver for each of the 8 tones. Run the loop once for each order. For the multiuser case with our definition of gap, each user's data rate is reduced by 1 bit on this complex baseband channel for 3 dB.

d. This is a simple use of the SWF function, which provides the two data rates that need to be converted to a loss.

e. This γ_{MAC} is for the energy-sum MAC, so use the macmax program with the water-fill output single-user rate relative to macmax's output. (The macmax program's linear-receiver output is not of interest in this problem.)

MAC Input Design (7.2) Problem 5.17 cascades upon Problem 5.16's channel to investigate channel input design.

a. The user energy constraints are each 64/9 so try using minPMAC on this channel from Problem 5.16. The admMAC also can be used directly for feasibility check.

b. The larger data rate follows the same process, but may lead to different result.

BC Design (7.3) Problem 5.18 follows 5.16 with the same channel and a fixed equal-energy/dimension input, focusing on design using the mu_bc program.

a. The ν is the same for dual channels. The expansion of input dimensions occurs on this channel. For a square channel, the number of channel input dimensions per symbol remains the same also because this is a function of the vector DMT. All the tones are duals and may have enlarged discrete modulators; but the actual channel input uses the same dimensionality. There are two transmit antennas for each user on this BC.

b. The dual enumerates for each tone the process of transpose and order reversal.

c. This becomes a looped (or multidimensional reshaped) use of the mac2bc program. The solution loops over the tone index, but uses the reshape command for the input H. Your Rxxb dimensionality here should match dimensional answers in Part a.

d. When this is correct, the bcmax program should provide the same sum rate as the macmax that was used in Problem 5.16.

e. This is really an enumeration, but make sure your precoder matrices are large enough dimensionally. The mu_bc program is really helpful here. These formatting commands may be helpful;

```
for n=1:8
GU(:,:,n)=[Gunb{1,n} ; Gunb{2,n}];
end
MSWMFU=zeros(4,1,8);
for n=1:8
MSWMFU(:,:,n)=[MSWMFunb{1,n} ; MSWMFunb{2,n}];
end
```

[**BC Input Design**] More mature BC Design addresses the BC's input optimization for a given data rate and/or energy vector. Problem 5.19 cascades upon Problems 5.16-18's using MAC input optimization with duality and the mac2bc program to find an optimum BC design for the same channel.

a. This problem uses minPMAC for a design and then maps through duality to the corresponding BC design. Thus dual BC and MAC designs should match on individual user's data rates.

b. You can use admMAC here and it provides a different viable design. You should describe and understand the differences.

c. The loss is the single-user max relative to the maximum energy-sum rate. You already computed this in an earlier problem and it should be the same.

d. The margin follows by successive uses of admMAC until the attempted increased-rate point is no longer feasible.

IC (5.20)

a. The iw_polite program is new this year, and previous solutions for this problem had a simple IW program written to solve just this problem. You may want to try to match them (not tested yet by instructor).

```
Es = [1 1];
eps = 1;
while eps > 1e-5
    g1 = G(1,1)/(1+G(1,2)*Es(2));
    g2 = G(2,2)/(1+G(2,1)*Es(1));
    [gs, idx]=sort([g1,g2],'descend');
    for Nuse = 2:-1:1
        K = (2 + sum(1./gs(1:Nuse)))/Nuse;
        Es_new = K-1./gs(1:Nuse);
        if Es_new(end) > 0
            break
        end
    end
    Es_new(idx) = Es_new;
    eps = max(abs(Es_new - Es));
    Es = Es_new;
end
bs = log2(1+Es.*[g1,g2]);
fprintf('E1 = %.4f, b1 = %.4f\nE2 = %.4f, b2 = %.4f\nrate sum = %.4f\n\n',...
    Es(1), bs(1), Es(2), bs(2), sum(bs));
```

b. Solution simplifies by noting that user 2 has no crosstalk. 12.28 and is better than the IW solution in Part a.