Homework Help - Problem Set 6
Solutions

[Singularity Removal] Singularity in the discrete-modulator context occurs when energy (non-zero signal) expands its dimensionality, basically from $\overline{N}^*$ to $\overline{N}$ or $\overline{N}+\nu$ or even to ∞. Essentially, the discrete modulator interpolates finite energy (power as $N \rightarrow \infty$ and normalized) from the smaller number of dimensions to the larger number. Decimation is the discrete demodulator's corresponding function. The singular dimensions are the ones carrying no energy (nor signal) but those extra dimensions are imposed by system architecture or field implementation/constraints. The only input components that make it to the output, though, are the ones that are nonsingular in both the input and then only those components' components that lay within the channel's pass space.

Appendix D's Paley-Wiener theorems are continuous-time and discrete-time tests for a signal or channel's singularity (if PWC is satisfied, it is nonsingular). The finite-length case PWC equivalent is a non-zero matrix determinant ($R_{\boldsymbol{x}\boldsymbol{x}}$ or R_f respectively for input and channel). GDFE theory calls attention to non-singularity and provides way to understand and transcend it

The HW6 problems that eliminates similarity simply rework the basic small matrix channel or input autocorrelation by following directly Figure 5.5's and 5.7's (the channel- and input-singularity flow charts respectively) 8-step processes. For any input autocorrelation matrix, the singularity tests is a zero determinant, $|R_{xx}| = 0$. A little better yet is using the eig command in matlab to find the eigenvalues. If they are all non-zero (and positive real or you have something that is not an autocorrelation matrix to start), then no singularity. Cyclic matrices are never singular (unless trivialy all zeros). Another type of matrix that is never singular is any autocorrelation PLUS I (often found in MMSE estimation and all real systems that have finite noise everywhere, e.g. nonzero noise on all dimensions).

There are infinite choices for a square root matrix $R_{\boldsymbol{x}\boldsymbol{x}}^{1/2}$ (whether singular or not), so there are many discrete-modulator choices. A common one uses the eigenvectors directly, equivalently singular vectors borrowed from a channel description like in vector coding, Also found are the Cholesky factors (or canonical factors), which are useful for certain implementations, particularly in multi-user situations or those with very low latency requirements. However, any square root $R_{\boldsymbol{x}\boldsymbol{x}}^{1/2}$ becomes another square root via $R_{\boldsymbol{x}\boldsymbol{x}}^{1/2}(2) \rightarrow R_{\boldsymbol{x}\boldsymbol{x}}^{1/2}(1) \cdot Q$ for any matrix for which $Q \cdot Q^* = I$. With complex systems (or any system with 2 or more real dimensions), there are an infinite number of possible square roots. Thus, as a designer or as a student doing homework, you have choices!.

[**Matrix Bias**] Repeated from earlier assignment.

Appendix D addresses Mean-Square Error (MSE), but basically as the name says, it is the mean of the difference between the two quantities, namely the error, squared. Thus if

$$a = b + e$$

and the error between a and b is $e = a - b$, then the MSE is simply $\mathbb{E}[e^2]$ if e is AWGN. This applies to the AWGN when $a = \boldsymbol{y}$ and $b = H \cdot \boldsymbol{x}$ and the error is the AWGN noise $\boldsymbol{n}$. The reverse direction of estimating $\boldsymbol{x}$ from $\boldsymbol{y}$, however, is closer to a receiver's symbol detection/estimation. This is especially true if $\boldsymbol{x}$ is a discrete random vector. The average error probability is well approximated by

$$P_e \approx N_e \cdot Q\left(\frac{d_{min}}{2\sigma}\right)$$

where d_{min} just computes the the difference between the the corresponding noise-free channel outputs, so

$$d_{min} = \min_{\boldsymbol{x}' \neq \boldsymbol{x} \in C_{\boldsymbol{x}}} \|H \cdot \boldsymbol{x} - H \cdot \boldsymbol{x}'\| \ .$$

When $H = I$, this is simply the distance between the closest two code words. The quantity N_e is the average number of nearest neighbors, which in most cases is closely approximated by the number of other codewords that may be at the minimum distance. Chapter 1 addresses this area if more information required. A binary constellation simply has d_{min} as the distance between the two points and $N_e = 1$. A 4PAM constellation has $\pm 1, \pm 3$ with distance $d_{min} = 2$ and an average of 1.5 nearest neighbors for each point, so $P_e \approx 1.5 Q(1/\sigma)$ (the approximation is exact in this example). So if the 4PAM transmission system had $\bar{\mathcal{E}}_x = 5$ and $\sigma^2 = .04$ with $H = 1$, then a 4PAM system would have $d/(2 \cdot \sigma) = 2/(2 \cdot .2) = 5$, and the symbol-error probability is then $P_e = 1.5 \cdot Q(5)$; in matlab:

```
>> 1.5*Q(5) =   4.2998e-07
```

The linear MMSE estimator of channel input, given channel input is always biased with non-zero noise. Even in the simplest case of $y = x + n$, it is possible to multiply y by a number less than 1 that will have a MMSE that reduces the signal x just enough so that the consequent simultaneous decrease of noise is beneficial. If the data signal has energy such that $SNR = \bar{\mathcal{E}}_x / \sigma^2$, this shrinkage of the channel output produces $(1 - 1/SNR) \cdot x_k$ and a ratio $\bar{\mathcal{E}}_x / MMSE = SNR + 1$. That apparently larger SNR misleads somewhat in that the estimate is biased. Bias removal (multiply by $(SNR + 1)/SNR$ eliminates the bias and increases the noise back to the original level. Clearly, the MMSE would not help the $x + n$ channel, but with more general channels the MMSE estimate can reduce substantially crosstalk and intersymbol interference; however the same bias/scaling-error occurs. This amount remains

$SNR_{mmse} = \bar{\mathcal{E}}_x / M\bar{M}SE$ (with $SNR = SNR_{mmse} - 1$ and the same scale-up removes the bias.

A detector designed for the original x has a non-zero-mean Gaussian noise, given the input x. A non-zero (conditional) mean causes the error-probability calculation to reduce the minimum distance by the bias amount. The 4 PAM example above has $SNR = 25$ (14 dB). The bias then causes

$$d_{min,bias} \to d_{min} \cdot (1 - 1/25) = \frac{24}{25} \cdot d_{min}$$

or equivalently the $d_{min}/2$ in the P_e formula's Q-function argument subtracts $d_{min} \cdot 1/25$ while the $\sqrt{MMSE}$ reduces to $\sqrt{24/25} \cdot \sigma$. The numerator reduction offsets the denominator reduction in the trivial scalar AWGN case. However, if there had been residual ISI added to the noise at this MMSE biased SNR of 25 dB, the minimum distance decreases by the same amount in bias removal, but the ISI would reduce presumably by a larger amount than the noise, and the improvement in distance to residual error would more than offset. So, removing the bias produces a better (lower) error probability, here corresponding to an SNR of 24, which still exceeds the value ocuring if some other non-MMSE estimate of x is instead used.. This holds in general - always remove the bias, it always improves. In more general cases, despite the slight reduction from $SNR_{mmse} \to SNR_{mmse} - 1$ to remove the bias, this new MMSE-assisted receiver will always perform at least as well as if there were no MMSE estimator.

[**Colored Inputs**] Problem 5.8's Colored inputs have non-diagonal autocorrelation matrices (thus "not white"). They often can be better than "white" inputs in that they match a set of independent messages' transmission to a channel. The essence (entropy) of the information though is always carried by a white component. It is this component that the GDFE finds and estimates (recursively with non-trivial feedback).

The steps in the channel-input singularity flow chart (Figure 5.5 in current text) is a blue print on how to work Problem 5.8. This problem goes beyond the blue print to investigate what the signals actually look like (for 8 easy ± 1 length-3 input sequences, or 2 PAM). With the process correctly implemented from the Figures, simple linear combinations of the original ± 1 values produces the discrete modulator signals along the transmit processing path. Use matlab, but it is not hard to find them. Yes, they look strange, but are the part in the pass space. How much energy lost to pass space? There is a very easy way to find this (and a hard way). The problem also explores an optimum detector for the pass-speace signal component, but then illustrates also how easily the same information rate could be attained by simply coding on the "white part" of the input.

[**White Inputs, asymptotic convergence**] Problem 5.9 follows 5.8 with a white input. The cyclic prefix forces a cyclic channel that has empty null space always (unless all zeros). Thus, if all dimensions energized equally (white input), they make it through. The GDFE reduces to the simpler yet CDFE in this case. However, that white input may not be the best input energy distribution (we all know by now that water-filling is the best).

This problem illustrates a simple $1+D$ channel (often called "Duobinary"). At small $\overline{N}$, it is easy to work, but frankly the SNR sucks compared to reasonably simply MMSE-DFE design (just use the program from Chapter 3, you need not understand all the theory there - although that theory is a twisted on what you see in Chapters 2 and 5). Basically for small number of feedforward taps (16 is on the border of small perhaps and a bit complex) and only 1 feedback tap, the maximum SNR is attained that (same non-water-fill-white-input) DMT would get. The problem does also illustrate that a simple reallocation of one unit of energy to the pass space helps a bit, but not quite the solution here either. However, by increasing the block size, the CDFE does also get the MMSE-DFE result.

The problem attempts to illustrate that the CDFE takes a pretty long block length to get to the MMSE-DFE result; however the CDFE is better than the MMSE-DFE on any finite-length packet. One then starts to wonder (hopefully) about the MMSE-DFE's use on finite-length packets. Ultimately, with the fast algorithms for DFT (that is FFT), one is drawn back to the conclusion that the canonical and optimum transmission system (DMT when guard period is small fraction) is probably the wisest choice. It took a lot of effort (and > 1000 students through previous versions of this course) to get the industry to appreciate and move to DMT, C-OFDM for wireless, etc. This problem attempts to illustrate that efficacy, nowadays more widely understood (but there are still some confused out there too!).

[**Cyclic Antennas**] This MIMO Antenna problem might better be named cyclic antennas.

Basically, the problem as is tests the ability to recognize the cyclic channel, even if it occurs in space time. It's pretty easy from there and simply requires finding water-fill (hardest part but only small number of variables - borrow commands from inside the dmtra.m program if you like). Recall that the M matrix for circulant channel is the DFT matrix. This also checks if you can find the right Cholesky factor (or use Yun's "loch.m" program ...).

Once through it, you might be thinking of any way to create cyclic channels in time space. Indeed, this is what cellular MIMO does attempt to do in that the space-time transmit matrices (think per-tone if you like, but usually the same across many tones) are chosen from a set of DFT matrices with different "repeating tone widths." (The receiver just sends back a "modulation-coding index" on which DFT the transmitter should use after the receiver learns the channel.) That would only be optimum if the channel is cyclic, but space-time is not cyclic like time-frequency, or may is it?

On this problem think of 3 antennas in a triangle at transmitter and at receiver. In a fairly symmetric environment, what might you expect the crosstalk to look like? Now look at the H in this problem. How about 4 antennas? Of course real environments would destroy the crosstalk "cyclic symmetry," but how far off? At least to date, the 3GPP standards seem to believe that is pretty close.

An indeed with lots of antennas and thus no need for feedback sections, it may well be good enough.