Solutions to Homework 2

Problem 1

- Computation of $Q(D)$

$$
\begin{aligned}
\sum_{k:\text{odd}} 2^{-\frac{|k|-1}{2}} D^k &= \sum_{k:\text{odd},k>0} 2^{-\frac{|k|-1}{2}} D^k + \sum_{k:\text{odd},k<0} 2^{-\frac{|k|-1}{2}} D^k \\
& \quad k:\text{odd}, k<0 = \{-2m-1 | m=0,1,...\} \Longrightarrow \frac{|k|-1}{2} = m \\
& \quad k:\text{odd}, k>0 = \{2n+1 | n=0,1,...\} \Longrightarrow \frac{|k|-1}{2} = n \\
&= \sum_{n=0}^{\infty} 2^{-n} D^{2n+1} + \sum_{m=0}^{\infty} 2^{-m} D^{-2m-1} \\
&= D \sum_{n=0}^{\infty} (\frac{D^2}{2})^n + \frac{1}{D} \sum_{m=0}^{\infty} (\frac{1}{2D^2})^m \\
&= \frac{D}{1-\frac{D^2}{2}} + \frac{1}{D(1-\frac{1}{2D^2})} \\
&= \frac{D}{1-\frac{D^2}{2}} + \frac{D^{-1}}{1-\frac{D^{-2}}{2}}
\end{aligned}
$$

$$
\begin{aligned}
\sum_{k:\text{even}} 2^{-\frac{|k|}{2}} D^k &= \sum_{k:\text{even},k\geq 0} 2^{-\frac{|k|}{2}} D^k + \sum_{k:\text{even},k<0} 2^{-\frac{|k|}{2}} D^k \\
& \quad k:\text{even}, k<0 = \{-2m | m=1,2,...\} \Longrightarrow \frac{|k|}{2} = m \\
& \quad k:\text{even}, k\geq 0 = \{2n | n=0,1,...\} \Longrightarrow \frac{|k|}{2} = n \\
&= \sum_{n=0}^{\infty} 2^{-n} D^{2n} + \sum_{m=1}^{\infty} 2^{-m} D^{-2m} \\
&= \sum_{n=0}^{\infty} (\frac{D^2}{2})^n + \sum_{m=1}^{\infty} (\frac{1}{2D^2})^m \\
&= \frac{1}{1-\frac{D^2}{2}} + \frac{\frac{1}{2D^2}}{1-\frac{1}{2D^2}} \\
&= \frac{1}{1-\frac{D^2}{2}} + \frac{\frac{D^{-2}}{2}}{1-\frac{D^{-2}}{2}}
\end{aligned}
$$

- Factorization of $Q(D)$

$$
\begin{aligned}
Q(D) &= \sum_{k\in\mathbb{Z}} 2^{-\frac{|k|-1}{2}} D^k = \left[\frac{D}{1-\frac{D^2}{2}} + \frac{D^{-1}}{1-\frac{D^{-2}}{2}}\right] + \frac{5}{3}\left[\frac{1}{1-\frac{D^2}{2}} + \frac{\frac{D^{-2}}{2}}{1-\frac{D^{-2}}{2}}\right] \\
&= \frac{\frac{5}{3}+D}{1-\frac{D^2}{2}} + \frac{\frac{5}{6D^2}+\frac{1}{D}}{1-\frac{1}{2D^2}} \\
&= \frac{\frac{5}{3}-\frac{5}{6D^2}+D-\frac{1}{2D}+\frac{5}{6D^2}-\frac{5}{12}+\frac{1}{D}-\frac{D}{2}}{(1-\frac{1}{2}D^2)(1-\frac{1}{2}D^{-2})} \\
&= \frac{\frac{5}{4}+\frac{1}{2}D+\frac{1}{2}D^{-1}}{(1-\frac{1}{2}D^2)(1-\frac{1}{2}D^{-2})} \\
&= \frac{(1+\frac{1}{2}D)(1+\frac{1}{2}D^{-1})}{(1-\frac{1}{2}D^2)(1-\frac{1}{2}D^{-2})} \\
&= F(D)F^*(D^{-*})
\end{aligned}
$$

- Whitening Filter $W(D) = \frac{\sqrt{N_0}}{F^*(D^{-*})}$
  We design $W(D)$ such that the filtered sequence is causal. For that purpose we choose $F^*(D^{-*})$ as the anti-causal part of $Q(D)$.
  $F^*(D^{-*}) = \frac{1+\frac{1}{2}D^{-1}}{1-\frac{1}{2}D^{-2}}$

  Proof:
  The remaining part $\sqrt{N_0}F(D)$ is causal because: The ROC is $|z| < \sqrt{2}$. Then the unit circle is inside the ROC (stable filter) and the origin is in the ROC (causal filter).

  $W(D) = \sqrt{N_0}\frac{1-\frac{1}{2}D^{-2}}{1+\frac{1}{2}D^{-1}}$

Problem 2

(a)

$$
\begin{aligned}
p(t) &= \varphi(t) * h(t) \\
&= \frac{1}{\sqrt{T}}\text{sinc}\left(\frac{t}{T}\right) * \left(\delta(t) - \frac{2}{5}\delta\left(t-T\right)\right) \\
&= \frac{1}{\sqrt{T}}\left(\text{sinc}\left(\frac{t}{T}\right) - \frac{2}{5}\text{sinc}\left(\frac{t}{T}-1\right)\right)
\end{aligned}
$$

(b)

$$
\begin{aligned}
||p||^2 &= \int_{-\infty}^{\infty} p2(t)dt \\
&= \frac{1}{T}\left(\int_{-\infty}^{\infty} \text{sinc}^2\left(\frac{t}{T}\right) - \frac{4}{5}\,\text{sinc}\left(\frac{t}{T}\right)\text{sinc}\left(\frac{t}{T}-1\right) + \frac{4}{25}\,\text{sinc}^2\left(\frac{t}{T}-1\right)dt\right) \\
&= \frac{1}{T}\left(T + \frac{4}{25}T\right) = \frac{29}{25}
\end{aligned}
$$

$$\tilde{\varphi}(t) = \frac{p(t)}{||p||} = \sqrt{\frac{25}{29T}}\left(\text{sinc}\left(\frac{t}{T}\right) - \frac{2}{5}\text{sinc}\left(\frac{t}{T} - 1\right)\right)$$

(c) We have $\mathbb{E}[Z(t)Z^*(\tau)] = N_0\delta(t-\tau)$ and $Z_n =< Z, \tilde{\varphi}_n >$. So we can compute

$$\begin{aligned}
\mathbb{E}[Z_n, Z^*_{n-k}] &= \mathbb{E}\left[\left\{\int Z(t)\tilde{\varphi}_n(t)dt\right\}\left\{\int Z^*(\tau)\tilde{\varphi}^*_{n-k}(\tau)d\tau\right\}\right] \\
&= \int\int \tilde{\varphi}_n(t)\tilde{\varphi}^*_{n-k}(\tau)\mathbb{E}[Z(t)Z^*(\tau)]dtd\tau \\
&= \int\int N_0\delta(t-\tau)\tilde{\varphi}_n(t)\tilde{\varphi}^*_{n-k}(\tau)dtd\tau \\
&= N_0\int \tilde{\varphi}_n(t)\tilde{\varphi}^*_{n-k}(t)dt \\
&= N_0\int \tilde{\varphi}(t-nT)\tilde{\varphi}^*(t-(n-k)T)dt \\
&= \frac{25N_0}{29T}\int \Bigg[\text{sinc}\left(\frac{t}{T}-n\right)\text{sinc}\left(\frac{t}{T}-(n-k)\right) \\
&\quad -\frac{2}{5}\text{sinc}\left(\frac{t}{T}-(n+1)\right)\text{sinc}\left(\frac{t}{T}-(n-k)\right) \\
&\quad -\frac{2}{5}\text{sinc}\left(\frac{t}{T}-n\right)\text{sinc}\left(\frac{t}{T}-(n-k+1)\right) \\
&\quad +\frac{4}{25}\text{sinc}\left(\frac{t}{T}-(n+1)\right)\text{sinc}\left(\frac{t}{T}-(n-k+1)\right)\Bigg]dt \\
\Rightarrow \quad & \mathbb{E}[z_n, z^*_{n-k}] = N_0 q_k \\
& \text{where } q_{-1} = -\frac{10}{29}, q_0 = 1, q_1 = -\frac{10}{29}
\end{aligned}$$

So $S_Z(D) = N_0 Q(D)$, where $Q(D)$ is the D-Transform of q_k.
The equivalent channel in the D domain is

$$Y(D) = \frac{1}{N_0}S_Z(D)||p||X(D) + Z(D)$$

To obtain a white noise, we must find a filter $G(D)$ such that the power spectral density of the noise is constant. In order to find such a filter, let's take the spectral factorization of $S_Z(D)$.

$$S_Z(D) = N_0 Q(D) = N_0(1 - \frac{10}{29}D - \frac{10}{29}D^{-1}) = F(D)F^*(D^{-*})$$

$$S_Z(D) = \sqrt{\frac{25N_0}{29}}(1 - \frac{2}{5}D)\sqrt{\frac{25N_0}{29}}(1 - \frac{2}{5}D^{-1})$$

And $F(D) = \sqrt{\frac{25N_0}{29}}(1 - \frac{2}{5}D)$. The whitening filter $G(D)$ is $\frac{\sqrt{N_0}}{F^*(D^{-*})} = \sqrt{\frac{29}{25}}\frac{1}{(1-\frac{2}{5}D^{-1})}$.
The resulting channel is

$$\tilde{Y}(D) = Y(D)G(D) = \frac{1}{\sqrt{N_0}}F(D)||p||X(D) + W(D)$$

where $W(D) = \frac{Z(D)\sqrt{N_0}}{F^*(D^{-*})}$ is white noise and $S_W(D) = N_0$. So

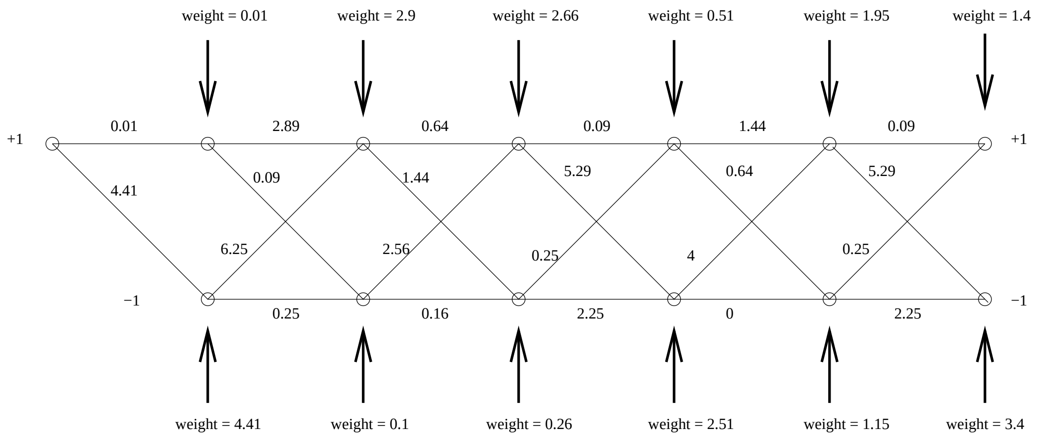
$$\tilde{Y}(D) = \sqrt{\frac{25}{29}}(1 - \frac{2}{5}D)\sqrt{\frac{29}{25}}X(D) + W(D)$$

$$\tilde{Y}(D) = (1 - \frac{2}{5}D)X(D) + W(D).$$

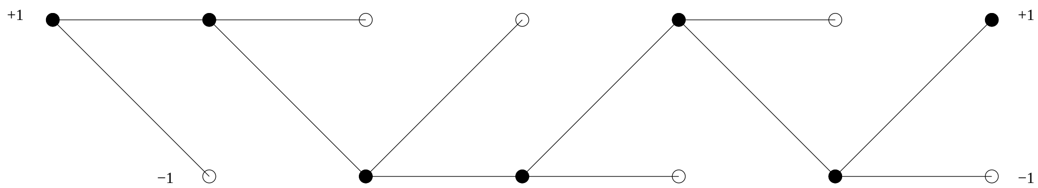
(d) The resulting channel in the time domain is

$$\tilde{y}_k = x_k - \frac{2}{5}x_{k-1} + w_k$$

The channel in the problem has a state of just X_{k-1} as $\nu = 1$. The Viterbi algorithm assumes a channel of the following form $y_k = u_k + z_k$ where $u_k = f_k * x_k$. In the considered channel $u_k = x_k - \frac{2}{5}x_{k-1}$ and $f_0 = 1$ and $f_1 = -\frac{2}{5}$.
Recall that the metric to use is $|y_k - u_k|^2$, and the maximum likelihood detection criterion is $\{x_k\} = \arg\min \sum_k |y_k - u_k|^2$
Shown below is the Viterbi algorithm decoding trellis.



Pruning the trellis gives us the sequence estimates. We used the fact that the initial and the end states are 1. Note that the choosing the end state to be 1 is a natural choice, because its accumulated weight is 1.4 and therefore smaller than the weight (3.4) of the -1 end state.



Hence the estimated input sequence is $\widehat{\mathbf{X}} = [+1, -1, -1, +1, -1, +1]$.

(e) The BCJR algorithm yields the following probability matrix (values are actual values / 10^{-4}).

			n			
$x(n)$	1	2	3	4	5	6
+1	2.5941	0.1168	0.2830	2.2811	0.7874	2.6177
-1	0.0237	2.5009	2.3348	0.3367	1.8304	0

So we see that the decoded sequence is $[+1, -1, +1, +1, -1, +1]$. We can also see the soft detection information the BCJR gives us; the larger the different between the values for $+1$ and -1 the more sure the estimate is. This decoded sequence doesn't agree with the sequence that we found for the Viterbi decoding. The reason is that the BCJR and MLSE maximize different metrics, so the decoded sequences can be different. The following Matlab code calculates the BCJR probabilities:

```
% Simple script to calculating the BCJR estimate of received values
% Assumes 1+D channel and X = +/- 1
% 2003: Created by Peter Berlin
% 2005: Inputs changed and comments added

%Received sequence
rec_y = [0.7 -1.1 -0.2 0.9 -0.6 0.9];
%Computation of uk accordind to all possible states
%trans = [(1)-0.4*(1); (-1)-0.4*(1); (1)-0.4*(-1); (-1)-0.4*(-1)]
trans = [0.6; 1.4; -1.4; -0.6];

len=length(rec_y);
dists = zeros(4,len);
for i = 1:len
    dists(:,i) = abs(rec_y(i) - trans);
end
dists;
N0 = 0.49
gamma = 0.5* normpdf(dists,0,sqrt(N0/2))


% We arrange things with the top row of alpha and beta correspond to
% S= +1
alpha = zeros(2,len);
alpha(:,1) = [1; 0];

for i = 2:len
    alpha(1,i) = alpha(1,i-1)*gamma(1,i-1) + alpha(2,i-1)*gamma(2,i-1);
    alpha(2,i) = alpha(1,i-1)*gamma(3,i-1) + alpha(2,i-1)*gamma(4,i-1);
end
alpha

beta = zeros(2,len+1);
beta(:,len+1) = [1; 0];

for i = len:-1:2
    beta(1,i) = beta(1,i+1)*gamma(1,i) + beta(2,i+1)*gamma(3,i);
    beta(2,i) = beta(1,i+1)*gamma(2,i) + beta(2,i+1)*gamma(4,i);
end
beta

symbol_probs = zeros(2,len);
for i = 1:len
    symbol_probs(1,i) = alpha(1,i)*gamma(1,i)*beta(1,i+1)
        + alpha(2,i)*gamma(2,i)*beta(1,i+1);
```

```
    symbol_probs(2,i) = alpha(1,i)*gamma(3,i)*beta(2,i+1)
        + alpha(2,i)*gamma(4,i)*beta(2,i+1);
end
symbol_probs

%Find the maxes now
[vals,indexes] = max(symbol_probs,[],1);
decoded_symbols = indexes;
decoded_symbols(decoded_symbols==2) = -1;
decoded_symbols
```

Problem 3

(a)

$$\begin{aligned}\mathbb{P}\big(X(n) = x_0 | \mathbf{Y} = \mathbf{y}\big) P_{\mathbf{Y}}(\mathbf{y}) &= P_{\mathbf{Y},X(n)}(\mathbf{y}, x_0) \\ &= P_{\mathbf{Y}|X(n)}(\mathbf{y}|x_0)\mathbb{P}(X(n) = x_0)\end{aligned}$$

(b) Assume that $H^{(n)}$ is an arbitrary decision rule for detecting $X(n)$. Let Γ_0 be the region such that $\forall \mathbf{y} \in \Gamma_0$, $H^{(n)}(\mathbf{y}) = x_0$ and similarly Γ_1 is the region associated to x_1.

For $\pi_0 = \mathbb{P}[X(n) = x_0]$ and $\pi_1 = \mathbb{P}[X(n) = x_1]$

$$\begin{aligned}\mathbb{P}[\text{error}] &= \mathbb{P}[H^{(n)}(\mathbf{y})\texttt{is wrong}] \\ &= \pi_0 \mathbb{P}[\mathbf{y} \in \Gamma_1 \mid X(n) = x_0] + \pi_1 \mathbb{P}[\mathbf{y} \in \Gamma_0 \mid X(n) = x_1] \\ &= \pi_0 \int_{\Gamma_1} P_{\mathbf{Y}|X(n)}(\mathbf{y} \mid x_0) d\mathbf{y} + \pi_1 \int_{\Gamma_0} P_{\mathbf{Y}|X(n)}(\mathbf{y} \mid x_1) d\mathbf{y} \\ &= \pi_0 \int_{\Gamma_1} P_{\mathbf{Y}|X(n)}(\mathbf{y} \mid x_0) d\mathbf{y} + \pi_1 \left[1 - \int_{\Gamma_1} P_{\mathbf{Y}|X(n)}(\mathbf{y} \mid x_1) d\mathbf{y}\right] \\ &= \pi_1 + \int_{\Gamma_1} \left[\pi_0 P_{\mathbf{Y}|X(n)}(\mathbf{y} \mid x_0) - \pi_1 P_{\mathbf{Y}|X(n)}(\mathbf{y} \mid x_1)\right] d\mathbf{y} \\ &= \pi_1 + \int_{\mathbb{R}^N} \underbrace{1_{\{\mathbf{y} \in \Gamma_1\}} \left[\pi_0 P_{\mathbf{Y}|X(n)}(\mathbf{y} \mid x_0) - \pi_1 P_{\mathbf{Y}|X(n)}(\mathbf{y} \mid x_1)\right] d\mathbf{y}}_{\textit{to make this term the smallest, collect all the negative area}}\end{aligned}$$

Therefore, in order to make the error probability smallest, we choose on $\mathbf{y} \in \Gamma_1$ if

$$\pi_0 P_{\mathbf{Y}|X(n)}(\mathbf{y} \mid x_0) \quad < \quad \pi_1 P_{\mathbf{Y}|X(n)}(\mathbf{y} \mid x_1)$$

That is, Γ_1 should be defined as,

$$\frac{\mathbb{P}[X(n) = x_0] P_{\mathbf{Y}|X}(\mathbf{y} \mid x_0)}{P_{\mathbf{Y}}(\mathbf{y})} \quad < \quad \frac{\mathbb{P}[X(n) = x_1] P_{\mathbf{Y}|X(n)}(\mathbf{y} \mid x_1)}{P_{\mathbf{Y}}(\mathbf{y})}$$

or $\mathbf{y} \in \Gamma_1$, if,

$$P_{X(n)|\mathbf{Y}}(x_0 \mid \mathbf{y}) \quad < \quad P_{X(n)|\mathbf{Y}}(x_1 \mid \mathbf{y}).$$

(c) Using these new priors:

$$
\begin{aligned}
P_e &= \tilde{\mathbb{P}}(x(n) = x_1)\mathbb{P}(H^{(n)}(\mathbf{Y}) = x_0|x(n) = x_1) \\
&\quad + \tilde{\mathbb{P}}(x(n) = x_0)\mathbb{P}(H^{(n)}(\mathbf{Y}) = x_1|x(n) = x_0)
\end{aligned}
$$

$$
\begin{aligned}
\min_{H^{(n)}} P_e &= \min_{H^{(n)}} \tilde{\mathbb{P}}(x(n) = x_1)\mathbb{P}(H^{(n)}(\mathbf{Y}) = x_0|x(n) = x_1) \\
&\quad + \tilde{\mathbb{P}}(x(n) = x_0)\mathbb{P}(H^{(n)}(\mathbf{Y}) = x_1|x(n) = x_0) \\
&= \min_{H^{(n)}} \frac{C_{01}\mathbb{P}(x(n) = x_1)}{\alpha}\mathbb{P}(H^{(n)}(\mathbf{Y}) = x_0|x(n) = x_1) \\
&\quad + \frac{C_{10}\mathbb{P}(x(n) = x_0)}{\alpha}\mathbb{P}(H^{(n)}(\mathbf{Y}) = x_1|x(n) = x_0) \\
&= \min_{H^{(n)}} C_{01}\mathbb{P}(x(n) = x_1)\mathbb{P}(H^{(n)}(\mathbf{Y}) = x_0|x(n) = x_1) \\
&\quad + C_{10}\mathbb{P}(x(n) = x_0)\mathbb{P}(H^{(n)}(\mathbf{Y}) = x_1|x(n) = x_0) \\
&= \min_{H^{(n)}} R(H^{(n)})
\end{aligned}
$$

(d) We will use the following notation:

$$
\bar{j} = \begin{cases} 1 & \text{if } j = 0 \\ 0 & \text{if } j = 1 \end{cases} .
$$

$$
\begin{aligned}
\hat{x} &= \arg\max_{j\in\{0,1\}} \tilde{\mathbb{P}}[X(n) = x_j|\mathbf{Y} = \mathbf{y}] \\
&= \arg\max_{j\in\{0,1\}} \frac{P_{\mathbf{Y}|X(n)}(\mathbf{y}|x_j)\tilde{\mathbb{P}}(x(n) = x_j)}{P_{\mathbf{Y}}(\mathbf{y})} \\
&= \arg\max_{j\in\{0,1\}} \frac{P_{\mathbf{Y}|X(n)}(\mathbf{y}|x_j)\frac{C_{\bar{j}j}\mathbb{P}(X(n)=x_j)}{\alpha}}{P_{\mathbf{Y}}(\mathbf{y})} \\
&= \arg\max_{j\in\{0,1\}} C_{\bar{j}j}\mathbb{P}(X(n) = x_j|\mathbf{Y} = \mathbf{y})) \\
&= \arg\max \{C_{10}\mathbb{P}(X(n) = x_0|\mathbf{Y} = \mathbf{y}); C_{01}\mathbb{P}(X(n) = x_1|\mathbf{Y} = \mathbf{y})\}
\end{aligned}
$$

(e) In question (d), we have derived the MAP rule for risk minimization. In class we have seen that the quantity

$$\mathbb{P}(X(n) = x_j|\mathbf{Y} = \mathbf{y})$$

can be computed using the α, β, γ recursion of the BCJR algorithm. This can still be done using the exact same algorithm. Only when we want to decide whether $\hat{X}(n)$ should be x_0 or x_1 we need to multiply these probabilities with the corresponding cost before comparing them.