EECE 5612 HW1
Stav Rones
2.2.2022

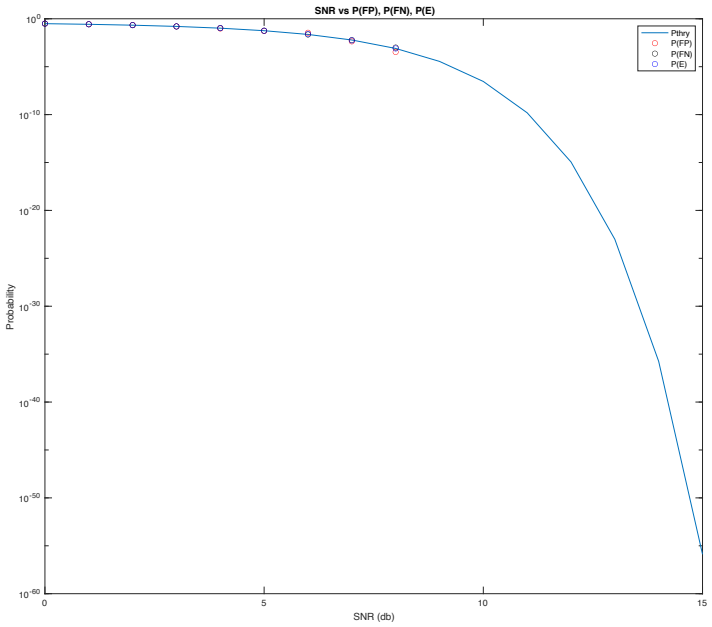
First, 0 mean noise with variance 1 was generated using randn(). The amplitude of the potential signal, A, was calculated from the given SNR using the following equation.

$$A = 10^{SNR_{db}/10}$$

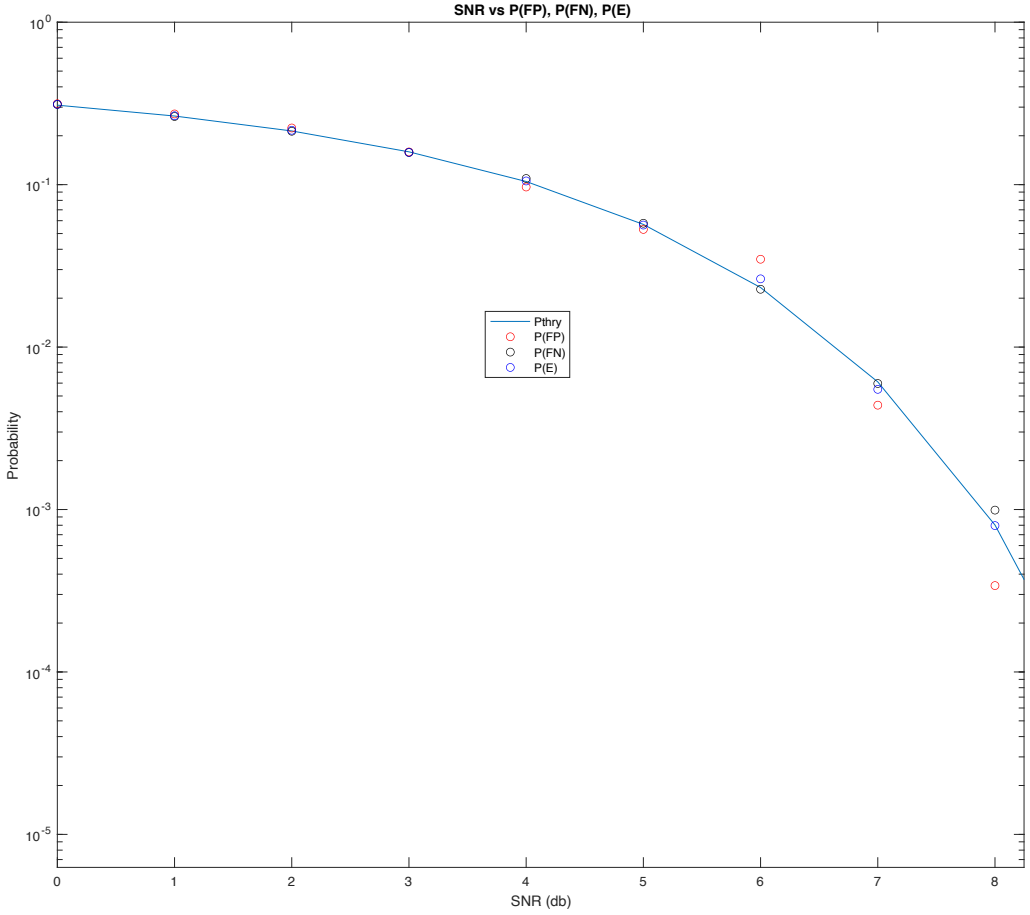
The signal was then generated by adding the amplitude to the noise with a probability of 0.7, which was achieved by comparing a uniform number generator, rand(), to 0.3. The ML decision rule was then applied, by comparing the signal to A / 2. The decision was then compared to the label to determine a false positive (FP) or false negative (FN). This process was performed 10,000 times, and the estimated values for P_{fp}, P_{fn}, and P_e were calculated using the following equations.

$$P_{fn} = \frac{FN}{TP + FN},\ P_{fp} = \frac{FP}{FP + TN}, P_e = P_{H_0}P_{fp} + P_{H_1}P_{fn}$$

This process was repeated 16 times for SNR values 0 to 15. The theoretical probability of error, $P_e = P_{fp} = P_{fn} = Q(\frac{A}{2})$, was plotted against the calculated values of P_{fp}, P_{fn} and P_e as shown in the following figure with a logarithmic y axis. Note that 0 values cannot be shown on a logarithmic axis.



The same plot is shown below zoomed in.



The behavior of the theoretical and calculated error probabilities are proportional and decreasing as SNR increases. This makes sense because as the SNR increases, the normal distribution of the signal value gets further away from the noise, making it much less likely for a signal to be part of the incorrect distribution. It is interesting to note that the P_e is closer to the P_{fn}, which make sense because the P_e considers the prior that the signal is more likely to occur. This means that there will be more false negatives than false positives since there are more opportunities for false negatives to occur than false positives.

APPENDIX

hw1.m

```
% EECE 5612 HW1
% Stav Rones
% 2.2.2022

function hw1

    % P_thry | Pfp | Pfn | Pe
    P = zeros(16,4);

    for SNR_db = 0:15
        i = SNR_db + 1;

        A = 10^((SNR_db)/10);

        % P Theroretical
        P(i,1) = qfunc(A/2);

        [P(i,2), P(i,3)] = mlTest10k(SNR_db);
        P(i,4) = 0.3 * P(i,2) + 0.7 * P(i,3);
    end

    figure(1)
    semilogy(0:15,P(:,1), 0:15, P(:,2), "ro" , 0:15, P(:,3), "ko", 0:15,
P(:,4), "bo")
    title('SNR vs P(FP), P(FN), P(E)')
    xlabel('SNR (db)')
    ylabel('Probability')
    legend("Pthry", "P(FP)", "P(FN)", "P(E)")
end


% Use ML to test binary hypotheses:
function [Pfp, Pfn] = mlTest10k(SNR)

    disp("Testing SNR = " + SNR);

    % Convert SNR to Amplitude
    A = 10^(SNR/10);

    FP = 0; FN = 0;
    TP = 0; TN = 0;

    for i = 1:10000

        % Generate signal
        y = randn();
        H0_label = true;
        if (rand() > 0.3)
            y = A + y;
            H0_label = false;
        end
```

```
        % Decide using ML
        H0_decision = true;
        if y > A / 2
            H0_decision = false;
        end


        if H0_label
            if H0_decision
                TN = TN + 1;
            else
                FP = FP + 1;
            end
        else
            if H0_decision
                FN = FN + 1;
            else
                TP = TP + 1;
            end
        end
    end

    Pfp = FP/(FP+TN);
    Pfn = FN/(FN+TP);
end
```