MEMO Number CMPE320_S21_PROJ4_CODE
DATE: 5/7/2021
TO: EFC LaBerge
FROM: Nem Negash
SUBJECT: MATLAB Code

1 MATLAB CODE

1.1 Design the MAP detector

```
function part2_1()
    %Plot Tau_map as a function of p0
    p_0 = (0.01:.01:0.99);
    %Sigma^2 = 1/ Gamma = 1/10
    var = 0.1;
    T_map = (var/2) .* log( ((1-p_0)) ./ p_0 );%Tau_map equation
    figure(1)
    plot(p_0, T_map,'Linewidth',2);
    title('T_{map}(p_{0})');
    xlabel('p_0');
    ylabel('T_{map}');
end
```

1.2 Investigate the MAP detector

```
function part2_2()
    A = 1;
    var = 0.1;%variance
    n = sqrt(var) * randn(1,100000);%noise
    R = -5:0.1:5;%x of plot
    x = rand(1,100000) >= 0.6;%1 or 0
    s = zeros(1,100000);
    %IID samples using x
    for i = 1:100000
        if x(i) <= 0.6
            s(i) = 1;
        else
            s(i) = -1;
        end
    end

    R_final = s + n;
    f_r0 = (((0.6)/(sqrt(2*pi*var))) * exp((-(R-A).^2)./(2*var)));
    f_r1 = (((0.4)/(sqrt(2*pi*var))) * (exp((-(R+A).^2)./(2*var))));
    hold on
    histogram(R_final, 'Normalization', 'pdf');
    plot(R,f_r0,'Linewidth', 3);
    plot(R,f_r1,'b', 'Linewidth', 3);

    title('Distribution of Received Signal');
    xlabel('R');
    ylabel('Probability');
```

```
    legend('Histogram','Positive A', 'Negative A');
end
```

1.3 Evaluate the ML Detector

```
function part2_3()
    g = [1:0.5:8, 8.5:0.25:13];%gamma
    var = 10.^(g/-10);%variance

    %getting 1 and 0
    bk = zeros(1,1000000);
    bk_old =  rand(1,1000000);
     for i = 1:100000
         bk(i) = bk_old(i);
        if bk(i) < 0.5
            bk(i) = 0;
        else
            bk(i) = 1;
        end
     end

    %mapping 1 and 0 to positive and negative
    m = zeros(1,1000000);
    for i = 1:1000000
        m(i) = bk(i);
        if m(i) == 0
            m(i) = -1;
        else
            m(i) = 1;
        end
    end

    %adding noise
    R = zeros(1,1000000);%recieved signal with noise
    errors = zeros(1,1000000);%errors
    pb_X = zeros(1,34);
    bk_hat = zeros(1,1000000);
    for i = 1:length(var)
        noise = sqrt(var(i)) * randn(1,1000000);
        for j = 1:1000000
            R(j) = m(j) + noise(j);
            bk_hat(j) = R(j);
            if bk_hat(j) >= 0
                bk_hat(j) = 0;
            else
                bk_hat(j) = 1;
            end
            errors(j) = xor(bk_hat(j), bk(j));
        end
        s = sum(errors(:)==0);
        pb_X(i) = s/1000000;
    end

    %generate analytical function
    analytical = 0.5*erfc(1./(sqrt(2*var)));
    figure(1)%plot using semilogy
```

```
    semilogy(g,analytical,'g', 'Linewidth', 3);%plot analytical
    hold on;
    semilogy(g,pb_X, '-or', 'Linewidth', 1.5);%plot simulation
    xlabel('Gamma');
    ylabel('Probability of Error');
    legend( 'Analytical','Simulation');
    title('ML Detector');
end
```

2.4 EVALUATE THE MAP DETECTOR

```
function part2_4()
    g = [1:0.5:8, 8.5:0.25:13];%gamma
    var = 10.^(g/-10);%variance
    tau = (log10(0.4/0.6).*var)/2;

    %getting 1 and 0
    bk = zeros(1,1000000);
    bk_old =  rand(1,1000000);
     for i = 1:100000
         bk(i) = bk_old(i);
        if bk(i) <= 0.6
            bk(i) = 0;
        else
            bk(i) = 1;
        end
     end

    %mapping 1 and 0 to positive and negative
    m = zeros(1,1000000);
    for i = 1:1000000
        m(i) = bk(i);
        if m(i) == 0
            m(i) = -1;
        else
            m(i) = 1;
        end
    end

    %add noise in order to get bk_hat
    r = zeros(1,1000000);%received singal
    e = zeros(1,1000000);%errors
    pb_X = zeros(1,34);
    bk_hat = zeros(1,1000000);
    for i = 1:length(var)
        noise = sqrt(var(i)) * randn(1,1000000);
        for j = 1:1000000
            if m(j) + noise(j) < tau(i)
                bk_hat(j) = 1;
            elseif m(j) + noise(j) >= 0
                bk_hat(j) = 0;
            end
            e(j) = xor( bk_hat(j),bk(j));
        end

        s = sum(e(:)== 0);
```

```
        pb_X(i) = s/1000000;
    end

    %generate analytical function
    top = 1-tau;
    top_1 = tau + 1;
    analytical = 0.3*erfc(top./(sqrt(2).*sqrt(var))) +
0.2*erfc(top_1./(sqrt(2).*sqrt(var)));
    figure(1)%plot using semilogy
    semilogy(g,analytical,'g', 'Linewidth', 3);%plot analytical
    hold on;
    semilogy(g,pb_X, '-or', 'Linewidth', 1.5);%plot simulation
    xlabel('Gamma');
    ylabel('Probability of Error');
    legend( 'Analytical','Simulation');
    title('MAP Detector');
end
```

1.5 Compare the MAP and ML Detector performance

```
%2.5 = 2.4/2.3
function part2_5()
    g = [1:0.5:8, 8.5:0.25:13];%gamma
    var = 10.^(g/-10);
    tau = (log10(0.4/0.6) .* var)./2;
    t_0_6 = 1 - tau;%p
    t_0_4 = 1 + tau;%1-p

    analytical_p0_50 = 0.5*erfc(1./(sqrt(2)*sqrt(var)));%2.3
    analytical_p0__60 = 0.3*erfc(t_0_6./(sqrt(2).*sqrt(var))) +
0.2*erfc(t_0_4./(sqrt(2).*sqrt(var)));%2.4

    figure(1)
    semilogy(g,analytical_p0__60,'g', 'Linewidth', 3);
    hold on
    semilogy(g,analytical_p0_50, '-oR', 'Linewidth', 1);
    xlabel('Gamma');
    ylabel('Probability of Error');
    title('Analytical at p_0 = 0.6 vs Analytical at p_0 = 0.5');
    legend('Analytical at p_{0} = 0.6','Analytical at p_{0} = 0.5');

    %rho function
    figure(2)
    p = analytical_p0__60 ./ analytical_p0_50;
    plot(g, p, 'Linewidth', 3);
    xlabel('Gamma');
    ylabel('Ratio');

    title('Ratio of p_0 = 0.6 / p_0 = 0.5 as a funtion of Gamma');
end
```