

VITERBI DECODING ALGORITHM

Github link to the jupyter notebook:

https://github.com/ashutosh-999/vtb_dcd_alg_mat.git

SCREENSHOTS

Generating the convolutional code:

```
for j = 1:IT
    for i = 1:N
        if i == 1
            u = 0;
            v = 0;
            w = data(j,1);

        elseif i == 2
            u = 0;
            v = data(j,1);
            w = data(j,2);

        else
            u = data(j,i-2);
            v = data(j,i-1);
            w = data(j,i);
        end

        y(i,1) = xor(u,v);
        y(i,2) = xor(xor(u,v),w);
        y(i,3) = xor(u,w);
    end
end
```

Calculating the Hamming distance:

```
for i = 1:N
    hd(i,1) = sum(xor(z(i,:),[0 0 0])); % 00 to 00 or state 0 to 0
    hd(i,2) = sum(xor(z(i,:),[0 0 1])); % 11 to 01 or state 3 to 1
    hd(i,3) = sum(xor(z(i,:),[0 1 0])); % 11 to 11 or state 3 to 3
    hd(i,4) = sum(xor(z(i,:),[0 1 1])); % 00 to 10 or state 0 to 2
    hd(i,5) = sum(xor(z(i,:),[1 0 0])); % 01 to 10 or state 1 to 2
    hd(i,6) = sum(xor(z(i,:),[1 0 1])); % 10 to 11 or state 2 to 3
    hd(i,7) = sum(xor(z(i,:),[1 1 0])); % 10 to 01 or state 2 to 1
    hd(i,8) = sum(xor(z(i,:),[1 1 1])); % 01 to 00 or state 1 to 0
end
```

Use of Dynamic Programming:

```
for i = 3:N
    h_err(i,1) = min(h_err(i-1,1) + hd(i,1), h_err(i-1,2) + hd(i,8));
    h_err(i,2) = min(h_err(i-1,3) + hd(i,7), h_err(i-1,4) + hd(i,2));
    h_err(i,3) = min(h_err(i-1,1) + hd(i,4), h_err(i-1,2) + hd(i,5));
    h_err(i,4) = min(h_err(i-1,3) + hd(i,6), h_err(i-1,4) + hd(i,3));

    if (h_err(i-1,1) + hd(i,1) < h_err(i-1,2) + hd(i,8))
        dp(i,1) = 0;
    else
        dp(i,1) = 1;
    end

    if (h_err(i-1,3) + hd(i,7) < h_err(i-1,4) + hd(i,2))
        dp(i,2) = 2;
    else
        dp(i,2) = 3;
    end

    if (h_err(i-1,1) + hd(i,4) < h_err(i-1,2) + hd(i,5))
        dp(i,3) = 0;
    else
        dp(i,3) = 1;
    end

    if (h_err(i-1,3) + hd(i,6) < h_err(i-1,4) + hd(i,3))
        dp(i,4) = 2;
    else
        dp(i,4) = 3;
    end
end
```

Plot of Bit Error Rate with the crossover probability(depends on noise level):

