
Table of Contents

Projet Partie II A23 ELE6701A	1
Génération des 1024 vecteurs hypothèses	1
Génération du bruit n	2
Génération de paquets s_i à envoyer	2
Vecteur y reçu	2
Paramètres pour calcul égaliseur MMSE	2
Egaliseur MMSE	2
Decision MMSE	3
Commentaire bii	6
Commentaire biii	6

Projet Partie II A23 ELE6701A

Bouh Abdillahi

Matricule : 1940646

```
clear all;
clc;
H=[1 0 0 0 0 0 0 0 0 0;
   -0.4 1 0 0 0 0 0 0 0 0;
   0 -0.4 1 0 0 0 0 0 0 0;
   0 0 -0.4 1 0 0 0 0 0 0;
   0 0 0 -0.4 1 0 0 0 0 0;
   0 0 0 0 -0.4 1 0 0 0 0;
   0 0 0 0 0 -0.4 1 0 0 0;
   0 0 0 0 0 0 -0.4 1 0 0;
   0 0 0 0 0 0 0 -0.4 1 0;
   0 0 0 0 0 0 0 0 -0.4 1];
```

Génération des 1024 vecteurs hypothèses

```
vecteurs_cell = cell(1, 1024);

for i = 1:1024
    % Convertir l'indice en binaire sur 10 bits
    binaire = dec2bin(i-1, 10) - '0'; % Convertir en vecteur binaire

    % Remplacer les '0' par '-1' dans la représentation binaire
    binaire(binaire==0) = -1;

    % Stocker le vecteur dans la cellule
    vecteurs_cell{i} = binaire';
end

% Convertir la cellule en une matrice de taille 10x1024
matrice_vecteurs_hypothese_s = cell2mat(vecteurs_cell);
```

Génération du bruit n

Variance désirée

```
variance_desiree = (0.15);

% Nombre de vecteurs à générer
nombre_paquets = 10000; %% VALEUR A MODIFIER
nombre_vecteurs = nombre_paquets;
taille_vecteur = 10;

% Génération des 1024 vecteurs de bruit gaussien blanc avec moyenne nulle
vecteurs_bruit_gaussien = sqrt(variance_desiree) *
    randn(taille_vecteur, nombre_vecteurs);
```

Génération de paquets s_i à envoyer

Nombre de vecteurs aléatoires à générer

```
nombre_vecteurs = nombre_paquets;
taille_vecteur = 10;

% Génération des vecteurs aléatoires
vecteurs_aleatoires_envoyes = randi([1, 2], taille_vecteur,
    nombre_vecteurs);
vecteurs_aleatoires_envoyes(vecteurs_aleatoires_envoyes == 2) = -1;
```

Vecteur y reçu

Matrice des vecteurs y

```
y=vecteurs_aleatoires_envoyes + vecteurs_bruit_gaussien;
```

Parametres pour calcul égaliseur MMSE

```
Rs = eye(3);
Rn = variance_desiree*Rs;
Delta=[0 1 2];

H=[1 -0.4 0 0;
    0 1 -0.4 0;
    0 0 1 -0.4];
eDelta1 = [1;0;0;0];
eDelta2 = [0;1;0;0];
eDelta3 = [0;0;1;0];
Ry = Rs*H*(H') + Rn;
```

Egalisateur MMSE

```
w_Delta1 = (eDelta1')*(H')*Ry^-1 % coefficients pour Delta = 0
```

```

w_Delta2 = (eDelta2')*(H')*Ry^-1 % coefficients pour Delta = 1
w_Delta3 = (eDelta3')*(H')*Ry^-1 % coefficients pour Delta = 2

MMSE_Delta1_theorique = 1 - w_Delta1*Ry*(w_Delta1')
MMSE_Delta2_theorique = 1 - w_Delta2*Ry*(w_Delta2')
MMSE_Delta3_theorique = 1 - w_Delta3*Ry*(w_Delta3')

w_Delta1 =

    0.8508    0.2865    0.0875

w_Delta2 =

   -0.0538    0.8237    0.2515

w_Delta3 =

   -0.0271   -0.0888    0.7362

MMSE_Delta1_theorique =

    0.1492

MMSE_Delta2_theorique =

    0.1547

MMSE_Delta3_theorique =

    0.2282

```

Decision MMSE

```

s_chapeau_iDelta1 = zeros(10,nombre_paquets);

s_chapeau_iDelta2 = zeros(10,nombre_paquets);
% s_chapeau_iDelta3 = zeros(10,nombre_paquets);

% vecteur_guess_s_Delta1 = zeros(10,nombre_paquets);
vecteur_guess_s_Delta2 = zeros(10,nombre_paquets);
% vecteur_guess_s_Delta3 = zeros(10,nombre_paquets);
error_count_symbol_bii = 0;
error_count_vector_bii = 0;
error_count_symbol_biii = 0;
error_count_vector_biii = 0;
for i=1:1:nombre_paquets

```

```

%      s_chapeau_iDelta1(:,i) = w1*[y(1,i); y(); y()];

      s_chapeau_iDelta2(:,i) = [w_Delta2*[y(2,i); y(1,i); 0];
                                w_Delta2*[y(3,i); y(2,i);
y(1,i)];
                                w_Delta2*[y(4,i); y(3,i);
y(2,i)];
                                w_Delta2*[y(5,i); y(4,i);
y(3,i)];
                                w_Delta2*[y(6,i); y(5,i);
y(4,i)];
                                w_Delta2*[y(7,i); y(6,i);
y(5,i)];
                                w_Delta2*[y(8,i); y(7,i);
y(6,i)];
                                w_Delta2*[y(9,i); y(8,i);
y(7,i)];
                                w_Delta2*[y(10,i);
y(9,i); y(8,i)];
                                w_Delta2*[0; y(10,i);
y(9,i)];
];

%      s_chapeau_iDelta3(:,i) = w3*y(:,i);

for j=1:1:10
%      if s_chapeau_iDelta1(j,i) < 0
%          vecteur_guess_s_Delta1(j,i) = -1;
%      else
%          vecteur_guess_s_Delta1(j,i) = 1;
%      end
%
%      if s_chapeau_iDelta2(j,i) < 0
%          vecteur_guess_s_Delta2(j,i) = -1;
%      else
%          vecteur_guess_s_Delta2(j,i) = 1;
%      end

%      if s_chapeau_iDelta3(j,i) < 0
%          vecteur_guess_s_Delta3(j,i) = -1;
%      else
%          vecteur_guess_s_Delta3(j,i) = 1;
%      end
end

% question bii
comp = isequal(vecteurs_aleatoires_envoyes(:,i),
vecteur_guess_s_Delta2(:,i));
difference_per_symbol_bii = nnz(vecteur_guess_s_Delta2(:,i) ~=
vecteurs_aleatoires_envoyes(:,i));
error_count_symbol_bii = error_count_symbol_bii +
difference_per_symbol_bii;
if comp
    test = 0;

```

```

        else
            error_count_vector_bii = error_count_vector_bii+1;
%            disp("error!");
        end

        % question biii
        difference_per_symbol_biii = nnz(y(:,i) ~=
vecteurs_aleatoires_envoyes(:,i));
        error_count_symbol_biii = error_count_symbol_biii +
difference_per_symbol_biii;
    end

    taux_erreur_par_paquet_de_10_symboles_bii = error_count_vector_bii/
nombre_paquets
    taux_erreur_par_symbole_bii = error_count_symbol_bii/
(nombre_paquets*10)

    diff_carree = (s_chapeau_iDelta2 - vecteurs_aleatoires_envoyes).^2;
    mmse_question_bi = mean(diff_carree(:))

    diff_carree = (y - vecteurs_aleatoires_envoyes).^2;
    mmse_question_biii = mean(diff_carree(:))

    taux_erreur_par_symbole_biii = error_count_symbol_biii/
(nombre_paquets*10)

    taux_erreur_par_paquet_de_10_symboles_bii =

        0.1969

    taux_erreur_par_symbole_bii =

        0.0215

    mmse_question_bi =

        0.2012

    mmse_question_biii =

        0.1506

    taux_erreur_par_symbole_biii =

        1

```

Commentaire bii

On remarque que l'erreur est de l'ordre de 2% pour chaque symbole. Cela représente une augmentation significative par rapport à la question I-d où nous avions de l'ordre de 0.4%

Commentaire biii

On remarque que le mmse en II-biii est plus faible qu'en II-bi. Par ailleurs on voit que le taux d'erreur par symbole est de 0.15% pour II-bii et est quasiment de 100% en II-biii.

Published with MATLAB® R2019b