

# Chaine de transmission d'image

## traitement d'image

Réalisé par :  
**Ait aamer mohamed**

UNIVERSITE SULTAN MOULAY SLIMANE  
FACULTE POLYDISCIPLINAIRE  
BENI-MELLAL

27 novembre 2023

# PLAN

- 1 Introduction
- 2 Type de chaine de transmission d'image
- 3 Chaine de transmission d'image analogique
- 4 Chaine de transmission d'image numerique
- 5 shemas du Chaine de transmission d'image numerique
- 6 Exemple des modulations/demodulations
- 7 Simulation sur Matlab
- 8 Conclusion

## Section 1

# Introduction

# Introduction

- À l'ère de la connectivité mondiale, la transmission d'images est devenue un pilier fondamental dans la manière dont nous partageons et recevons des informations visuelles. Ce projet s'attaque à l'optimisation de la chaîne de transmission d'images, une composante essentielle de nombreuses applications contemporaines, allant de la vidéoconférence à la surveillance en temps réel.

## Section 2

# Type de chaine de transmission d'image

# Type de chaine de transmission d'image

## Analogique

- Image est une source analogique
- La transmission d'une image analogique implique le transfert d'une représentation continue de l'image, souvent sous forme de signaux électriques ou optiques, plutôt que de données numériques discrètes.

# Type de chaine de transmission d'image

## Numerique

- Image est une source analogique + echantillonnage et quantification
- La transmission d'une image numérique implique le transfert d'une représentation discrète de l'image, où l'information visuelle est convertie en données numériques..

## Section 3

# Chaîne de transmission d'image analogique



# Chaîne de transmission d'image analogique

## Acquisition de l'image

- L'image est capturée à l'aide d'un dispositif tel qu'une caméra analogique ou un scanner. Ces dispositifs convertissent la scène visuelle en un signal électrique continu qui représente l'intensité lumineuse à différents points de l'image.

## Prétraitement (en option)

- Certains systèmes peuvent inclure des étapes de prétraitement pour améliorer la qualité de l'image ou ajuster certains paramètres. Cela peut inclure des filtres, des corrections de couleur, etc.

# Chaîne de transmission d'image analogique

## Modulation du Signal

- Le signal analogique, généralement sous forme de tension électrique, est modulé pour être compatible avec le moyen de transmission choisi. La modulation peut prendre différentes formes, comme l'amplitude, la fréquence ou la phase modulation, en fonction du support de transmission (câbles, ondes radio, etc.).

## Transmission

- Le signal modulé est transmis sur un support de transmission. Cela peut être réalisé via des câbles (par exemple, un câble coaxial) ou des signaux sans fil (par exemple, transmission radiofréquence).

# Chaîne de transmission d'image analogique

## Réception

- Le signal est reçu à l'autre extrémité de la transmission. Une étape de démodulation est effectuée pour extraire le signal analogique original à partir du signal modulé. Des dispositifs tels que des antennes, des amplificateurs et des filtres peuvent être utilisés pour améliorer la qualité du signal reçu.

## Affichage ou Stockage

- Une fois le signal analogique reconstitué, il peut être converti en une image visible à l'aide d'un écran ou d'autres dispositifs d'affichage. Dans certains cas, l'image analogique peut également être stockée sur des supports analogiques comme des cassettes vidéo.

## Section 4

# Chaine de transmission d'image numerique

# Chaîne de transmission d'image numérique

## Acquisition de l'image

- L'image est capturée à l'aide d'un dispositif numérique tel qu'un appareil photo numérique, une caméra de surveillance, ou un scanner. Ces dispositifs convertissent la scène visuelle en données numériques en utilisant des capteurs d'image.

## numérisation

- Si l'image n'est pas déjà sous forme numérique, elle est numérisée à l'aide d'un convertisseur analogique-numérique (CAN). Ce processus attribue une valeur numérique à chaque pixel de l'image, représentant son niveau d'intensité lumineuse.

# Chaîne de transmission d'image numérique

## compression

- Les données numériques peuvent être compressées pour réduire leur taille avant la transmission. La compression réduit la quantité d'informations nécessaires pour représenter l'image tout en essayant de conserver la qualité visuelle. Des algorithmes de compression tels que JPEG, PNG, ou d'autres peuvent être utilisés.

## Transmission

- Les données numériques de l'image sont transmises sur un support de transmission, tel qu'un réseau informatique (par exemple, Internet), des câbles (par exemple, HDMI, USB) ou des systèmes sans fil (par exemple, Wi-Fi, Bluetooth). Les protocoles de transmission, tels que TCP/IP, assurent la fiabilité et la gestion des erreurs lors de la transmission.

# Chaîne de transmission d'image numérique

## Réception

- Les données numériques sont reçues à l'autre extrémité de la transmission.

## decompression

- processus de décompression peut être appliqué pour restaurer les données à leur état d'origine. Les données sont ensuite converties en une image visible à l'aide d'un dispositif d'affichage.

## Affichage ou Stockage

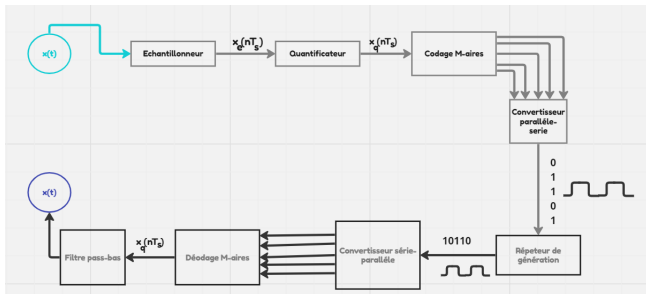
- L'image numérique est affichée sur un écran ou tout autre dispositif de visualisation.

## Section 5

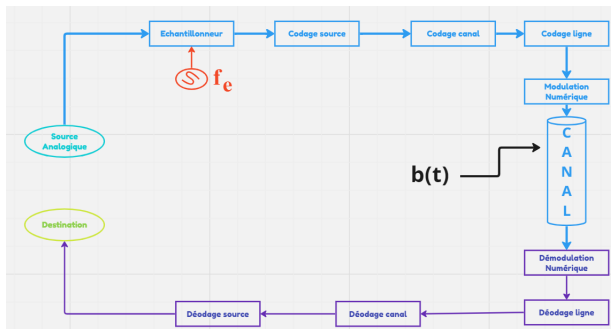
# shemas du Chaine de transmission d'image numerique



## Exemple :



## Exemple :

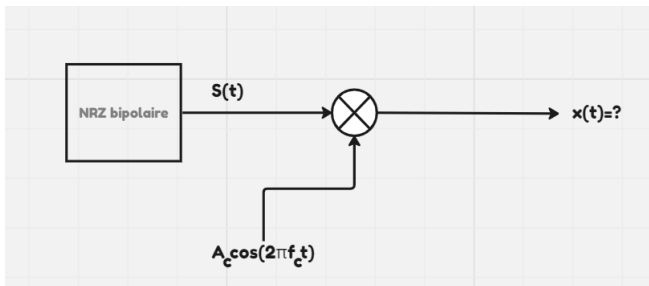


## Section 6

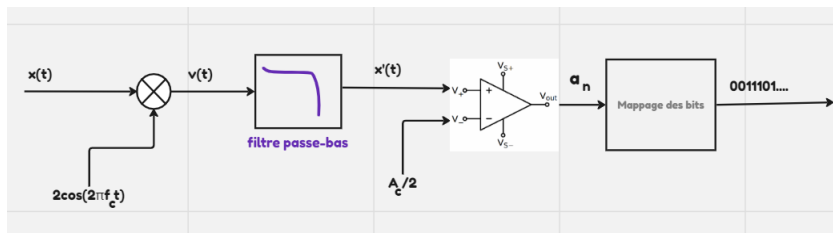
# Exemple des modulations/demodulations

# Exemples des modulations

- Représentation de modulation BPSK :



- Représentation de demodulation :



## Section 7

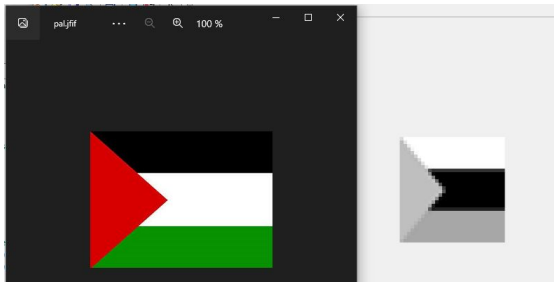
# Simulation sur Matlab

# Simulation sur Matlab

```
function binary_sequence = imageToBinary(image_path)
% Charger une image
image = imread(image_path);
imshow(image)

% Convertir l'image en niveaux de gris si elle est en couleur
if size(image, 3) == 3
    image = rgb2gray(image);
    image = imresize(image, [30 30]);
    img = imcomplement(image);
    imshow(img)
end

% Convertir l'image en une suite binaire
binary_sequence = reshape(dec2bin(img(:), 8).-'0', 1, []);
% binary_sequence = reshape(img, 1, 900);
end
```



Columns 781 through 793													
255	255	255	255	255	255	255	255	253	50	0	0	0	0
Columns 794 through 806													
0	0	0	0	0	0	0	34	167	167	167	167	167	167
Columns 807 through 819													
167	167	167	167	255	255	255	255	255	255	255	255	255	253
Columns 820 through 832													
50	0	0	0	0	0	0	0	0	0	0	34	167	
Columns 833 through 845													
167	167	167	167	167	167	167	167	255	255	255	255	255	
Columns 846 through 858													
255	255	255	253	50	0	0	0	0	0	0	0	0	0
Columns 859 through 871													
0	0	34	167	167	167	167	167	167	167	167	167	167	255
0	0	0	0	0	0	0	0	0	0	0	0	0	0
Columns 5 188 through 5 200													
0	0	0	0	0	0	0	0	0	0	0	0	0	0
Columns 5 201 through 5 213													
0	0	1	0	0	0	1	0	1	0	1	0	0	
Columns 5 214 through 5 226													
1	1	1	1	0	1	0	0	1	1	1	1	1	0
Columns 5 227 through 5 239													
1	0	0	1	1	1	1	0	1	0	0	1	1	
Columns 5 240 through 5 252													
1	1	0	1	0	0	1	1	1	1	0	1	0	
Columns 5 253 through 5 265													
0	1	1	1	1	0	1	0	0	1	1	1	1	1



```
function Ham_seq = H_encode(data_bloc)
    n = 7; % longueur de mot de code
    k = 4; % longueur de code
    A = [ 1 1 1;1 1 0;1 0 1;0 1 1 ]; % matrice de controle
    G = [ eye(k) A ]; % matrice generatrice
    Ham_seq = mod(data_bloc*G,2);
```

```
>> data = [1 0 0 1];
>> seq = H_encode(data)
```

```
seq =
```

```
1      0      0      1      1      0      0
```

```
function [t,x] = nrz(bits, e)

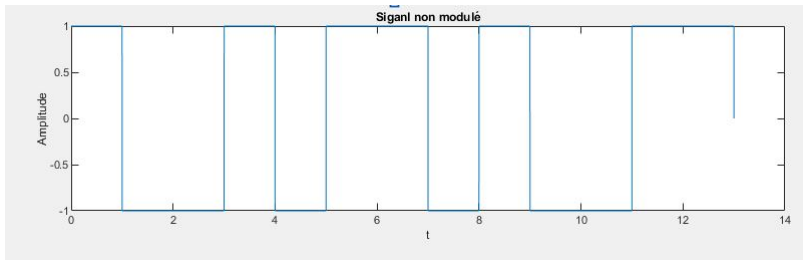
T = length(bits)/e; % longueur de tout la sequence de bits
                    % e =Nbr de bits par seconds
                    % bits = sequence de bits

n = 1000;
N = n*length(bits);
dt = T/N;
t = 0:dt:T;
x = zeros(1,length(t)); % signal de sortie

for i = 0:length(bits)-1
    if bits(i+1) == 1
        x(i*n+1:(i+1)*n) = 1;
    else
        x(i*n+1:(i+1)*n) = -1;
    end
end

>> data = [1 0 0 1 0 1 1 0 1 0 0 1 1];
>> e = 1;
[t,s] = nrz(data,e);
figure;
subplot(4,1,1);
plot(t,s);
xlabel('t');
ylabel('Amplitude');
title('Siganl non modul ')
```

- Représentation graphique :

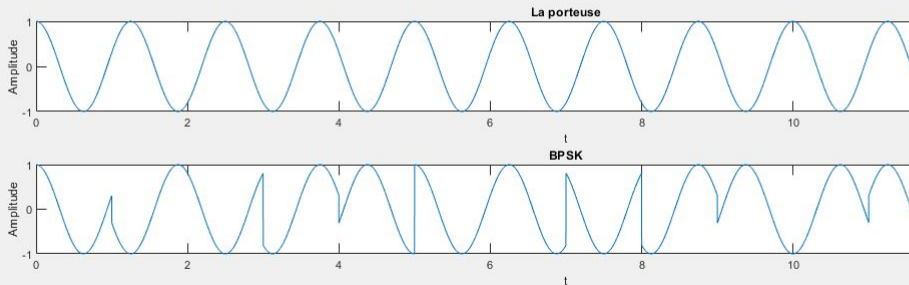


## ● Modulation :

```
***** %Modulation : *****  
Tb=0.1;  
fc=0.8; %frequence de la porteuse  
V=1;  
A=1;  
x=A.*cos(2*pi*fc*t); %la porteuse  
BPSK_signal=x.*s; % traçage du signal modulé avec "s" est le signal NRZ
```

```
>> data = [1 0 0 1 0 1 1 0 1 0 0 1 1];  
>> e = 1;  
[t,s] = nrz(data,e);  
>> Tb=0.1;  
fc=0.8;  
V=1;  
A=1;  
x=A.*cos(2*pi*fc*t);  
BPSK_signal=x.*s;  
>> subplot(4,1,2);  
plot(t,x);  
xlabel('t');  
ylabel('Amplitude');  
title('La porteuse')  
  
subplot(4,1,3);  
plot(t,BPSK_signal);  
xlabel('t');  
ylabel('Amplitude');  
title('BPSK');
```

- Représentation graphique :

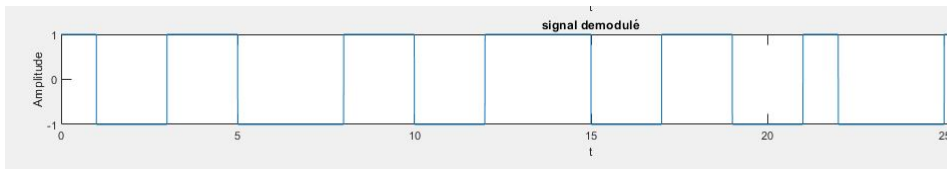


## • Demodulation :

```
%***** Demodulation : *****

BPSK_signal_demo = BPSK_signal.*x ;
for i=0:length(BPSK_signal_demo)-1 %Reconstruction du signal
    if BPSK_signal_demo(i+1)<=0 %Comparaison au seuil (zeros) et regene
        mm(i+1)=-1;
    else
        mm(i+1)=1;
    end
end
~
>> BPSK_signal_demo = BPSK_signal.*x ;
for i=0:length(BPSK_signal_demo)-1
    if BPSK_signal_demo(i+1)<=0
        mm(i+1)=-1;
    else
        mm(i+1)=1;
    end
end
>> subplot(4,1,4);
plot(t,mm);
xlabel('t');
ylabel('Amplitude');
title('signal demodulé');
```

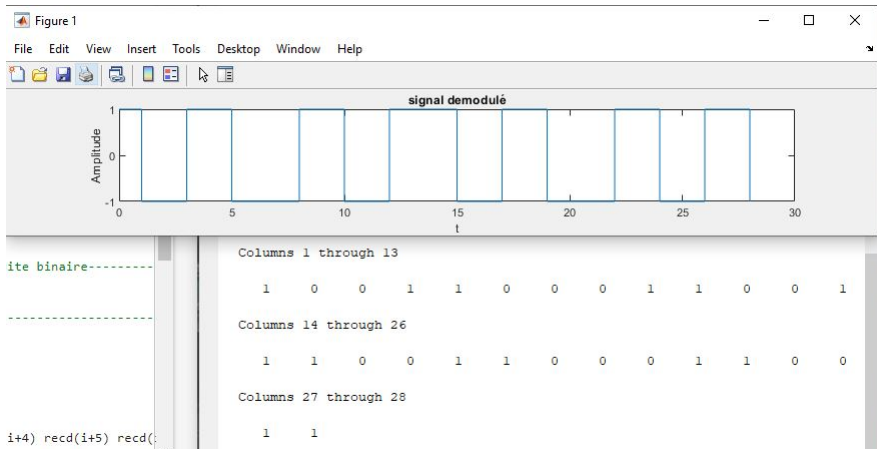
- Représentation graphique :



- Codage :

```
function binary = nrz_bipolar_to_binary(t,x)
    n = 1000;
    % signal est un vecteur de nombres réels représentant un signal NRZ bipolaire
    binary = []; % variable pour stocker la suite binaire
    for i = 1:n:length(t) % parcourir le vecteur du signal
        if x(i) == 1 % cas du premier élément
            binary = [binary 1]; % on ajoute un 1 arbitrairement
        else % cas des éléments suivants
            binary = [binary 0]; % on ajoute un 0
        end
    end
end
end
```





```
function Clair_seq = H_decode(recd)
    n = 7;
    k = 4;
    A = [ 1 1 1; 1 1 0; 1 0 1; 0 1 1 ];
    H = [ A' eye(n-k) ];
    % DECODER%
    syndrome = mod(recd * H',2);
    %Find position of the error in codeword (index)
    find = 0;
    for ii = 1:n
        if ~find
            errvect = zeros(1,n);
            errvect(ii) = 1;
            search = mod(errvect * H',2);
            if search == syndrome
                find = 1;
                index = ii;
            end
        end
    end
    disp(['Position of error in codeword=',num2str(index)]);
    correctedcode = recd;
    correctedcode(index) = mod(recd(index)+1,2); %Corrected codeword
    %Strip off parity bits
    msg_decoded=correctedcode;
    Clair_seq = msg_decoded(1:4);
```

```
>> recd_err = [1 0 1 1 1 0 0];  
>> seq = H_decode(recd_err);  
Position of error in codeword=3  
>> seq = H_decode(recd_err)  
Position of error in codeword=3  
  
seq =
```

```
1    0    0    1
```

## ● Decodage Hamming :

```
%-----decodeur du Hamming-----
R = length(recd);
clair = [];
if recd(R) == 0
    recd(R) = [];
end
L = length(recd);
for i = 1:7:L
    bloc = [recd(i) recd(i+1) recd(i+2) recd(i+3) recd(i+4) recd(i+5) recd(i+6)];
    seq = H_decode(bloc);
    clair = [clair seq];
end

>> recd = [1 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1];
>> R = length(recd);
clair = [];
if recd(R) == 0
    recd(R) = [];
end
L = length(recd);
for i = 1:7:L
    bloc = [recd(i) recd(i+1) recd(i+2) recd(i+3) recd(i+4) recd(i+5) recd(i+6)];
    seq = H_decode(bloc);
    clair = [clair seq];
end
```

## ● Sequence obtenue :

bloc =

1 1 0 1 1 0 0

Position of error in codeword=2

bloc =

0 1 0 0 0 1 1

Position of error in codeword=3

bloc =

1 1 0 1 1 0 0

Position of error in codeword=2

bloc =

0 1 0 0 0 1 1

Position of error in codeword=3

Position of error in codeword=2

seq =

1 0 0 1

Position of error in codeword=3

seq =

---

Position of error in codeword=2

clair =

1 0 0 1

Position of error in codeword=3

clair =

1 0 0 1 0 1 1 0

Position of error in codeword=2

clair =

1 0 0 1 0 1 1 0 1 0 0 1

Position of error in codeword=3

clair =

Columns 1 through 13

1 0 0 1 0 1 1 0 1 0 0 1 0

Columns 14 through 16

1 1 0

---

```
>> clair = [1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0];
```

```
>> r = sqrt(length(clair))
```

```
r =
```

```
4
```

```
>> M = reshape(clair,r,r);
```

```
>> M
```

```
M =
```

1	0	1	0
0	1	0	1
0	1	0	1
1	0	1	0

```
>> M'
```

```
ans =
```

1	0	0	1
0	1	1	0
1	0	0	1
0	1	1	0

```
>> imshow(M')
```

```
>> |
```

# Conclusion

En résumé, L'intégration croissante de l'intelligence artificielle dans les chaînes de transmission d'image ouvre la porte à des fonctionnalités adaptatives, basées sur l'apprentissage automatique, améliorant ainsi la qualité en fonction des conditions du réseau. Parallèlement, la sécurité et la confidentialité demeurent des préoccupations majeures, exigeant des solutions robustes pour protéger les données sensibles. En embrassant des principes d'interopérabilité, de réalité virtuelle et augmentée, d'efficacité énergétique, et en exploitant l'informatique périphérique, les futures chaînes de transmission d'image visent à redéfinir l'expérience visuelle à travers des environnements toujours plus diversifiés et exigeants.



FIN...