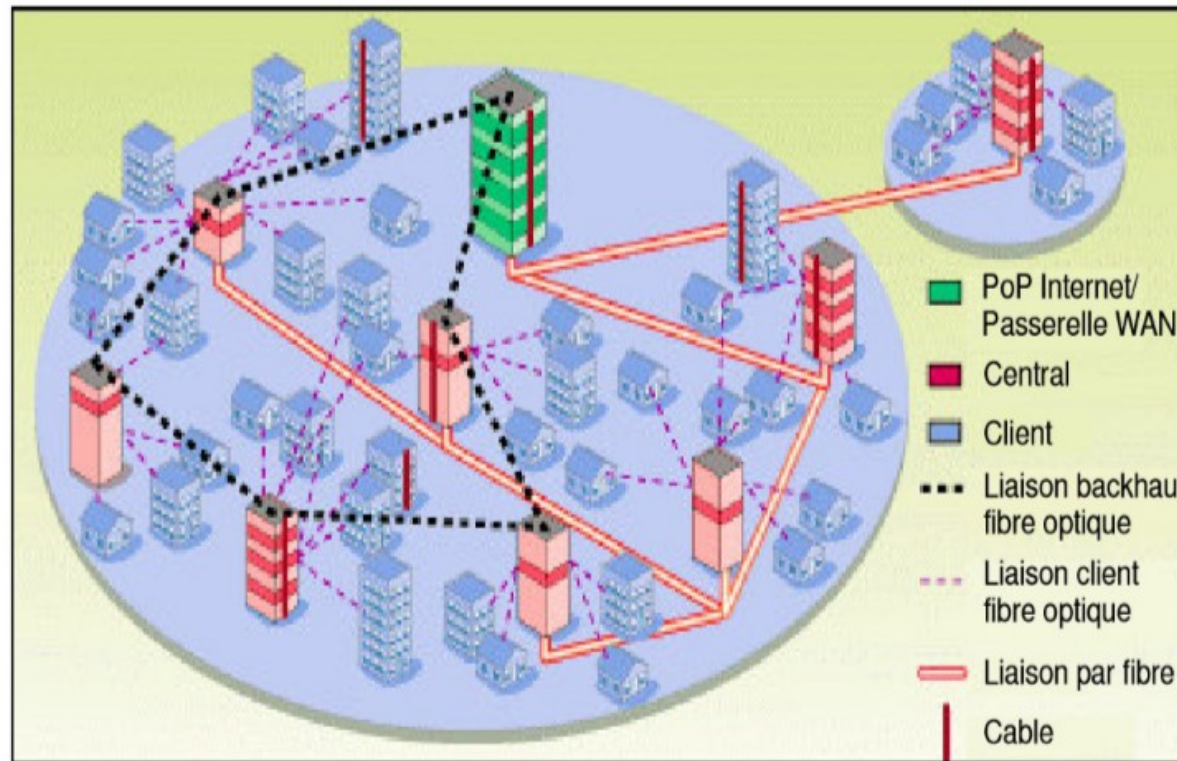


Transmission optique d'information dans l'atmosphère terrestre



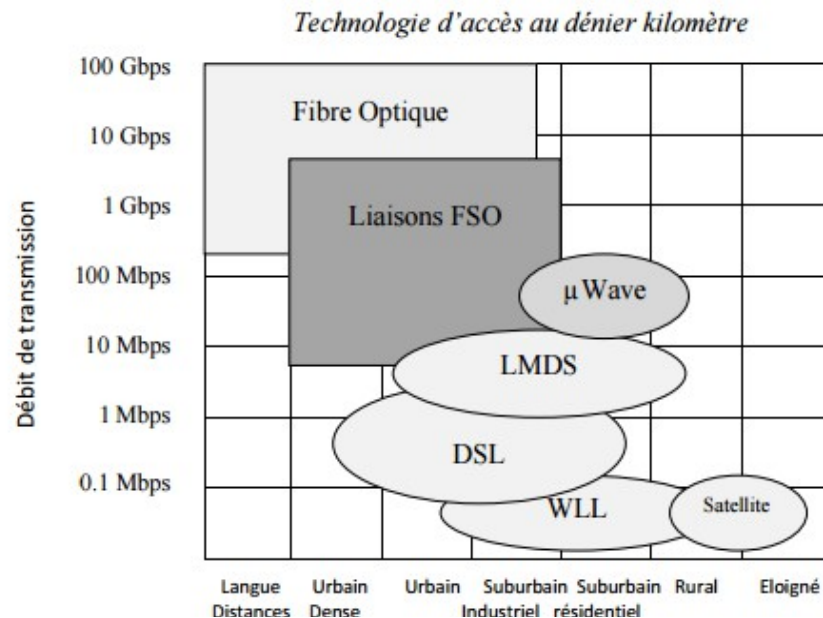
Principe de fonctionnement

- Liaison optique atmosphérique (LOA)
- Utilise un faisceau laser modulé pour transmettre une information



Avantages

- Faible distance :
 - peu coûteuse
 - facile d'installation
 - rapide
- Débit binaire équivalent à la fibre optique



Problématique physique d'une LOA

-La longueur d'onde de la transmission

-Atténuation du signal :

- Distance de liaison
- Divergence du faisceau
- Composition de l'atmosphère



phénomène d'absorption et de diffusion

Influence retenu

- Influence de la longueur d'onde



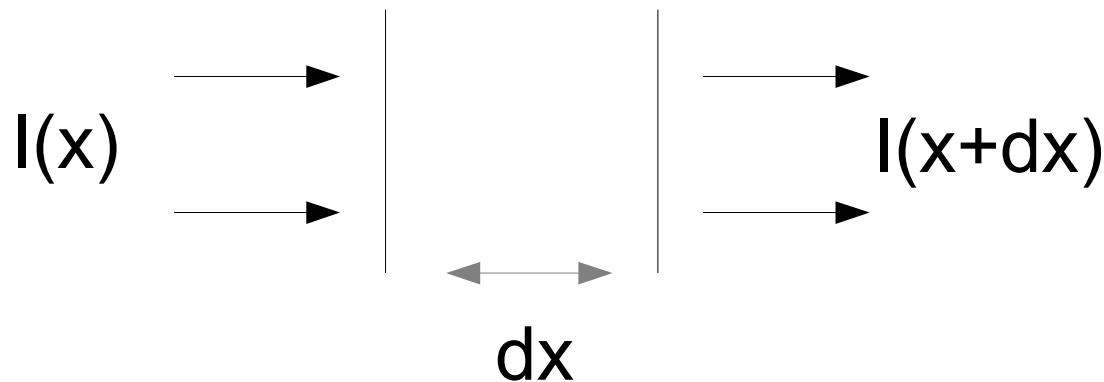
atténuation en fonction de la visibilité

- Différents types de modulations



modulation OOK, PPM, MPPM

Absorption moléculaire



$I(\lambda, x) - I(\lambda, x+dx) = dI_a(\lambda, x)$: intensité lumineuse absorbée

$$dI_a(\lambda, x) = \alpha(\lambda, x) I(\lambda, x) dx$$

↳ représente l'absorption du milieu

Transmission en milieu homogène :

$$T_a(\lambda, X) = \frac{I(\lambda, X)}{I(\lambda, 0)} = e^{-\alpha(\lambda) X}$$

Prédiction de l'atténuation optique d'ans l'atmosphère

- Définitions : •Visibilité $V(m) - 550nm$
 - Contraste C

-Par la loi de Koschmieder :

$$\frac{C(z=V)}{C(0)} = e^{-\beta(\lambda)V} = 0,05$$

β_v :coefficient de diffusion moléculaire/aérosolaire

$$\beta_v(\lambda) = \frac{\ln(20)}{V}$$

Modèle de Kruse

-En se basant sur les travaux empirique de Kruse :

$$\beta(\lambda)=\beta_v(\lambda)= \frac{-\ln(0,05)}{V} \left(\frac{\lambda}{0,55} \right)^{-q}$$

$$q = 1,6 \text{ si } V > 50\text{km}$$

$$1,3 \text{ si } 6\text{km} < V < 50\text{km}$$

$$0,585 V^{(1/3)} \text{ si } V < 6\text{km}$$

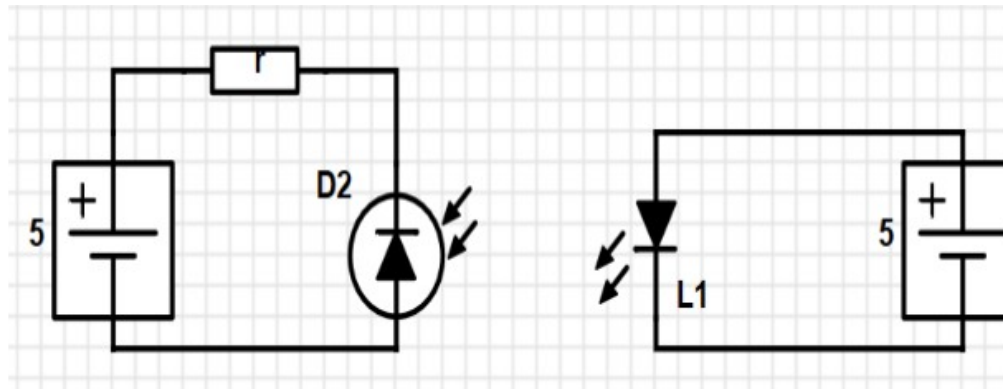
Conclusion : plus λ augmente, plus l'atténuation diminue

Étude expérimental

-Matériels utilisé :

- Émetteur :diode laser (480,550,650nm)
- Récepteur :photodiode (polarisé)
- Machine à fumer

-Schéma électrique :



Récepteur

-Loi de Malus : vérifier que le flux lumineux est proportionnel à l'intensité électrique

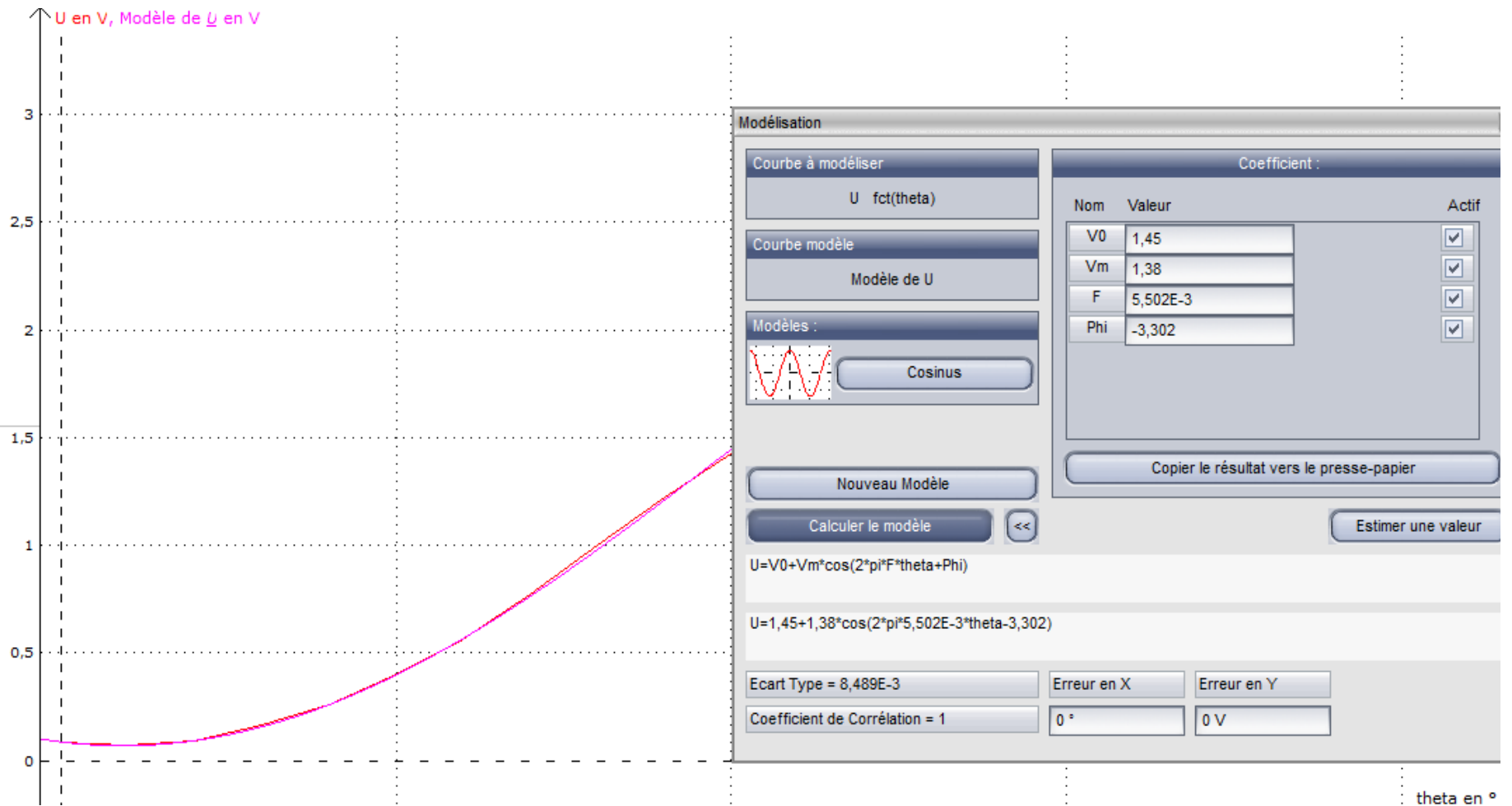


$$I_p = I_v \cos^2(\theta)$$

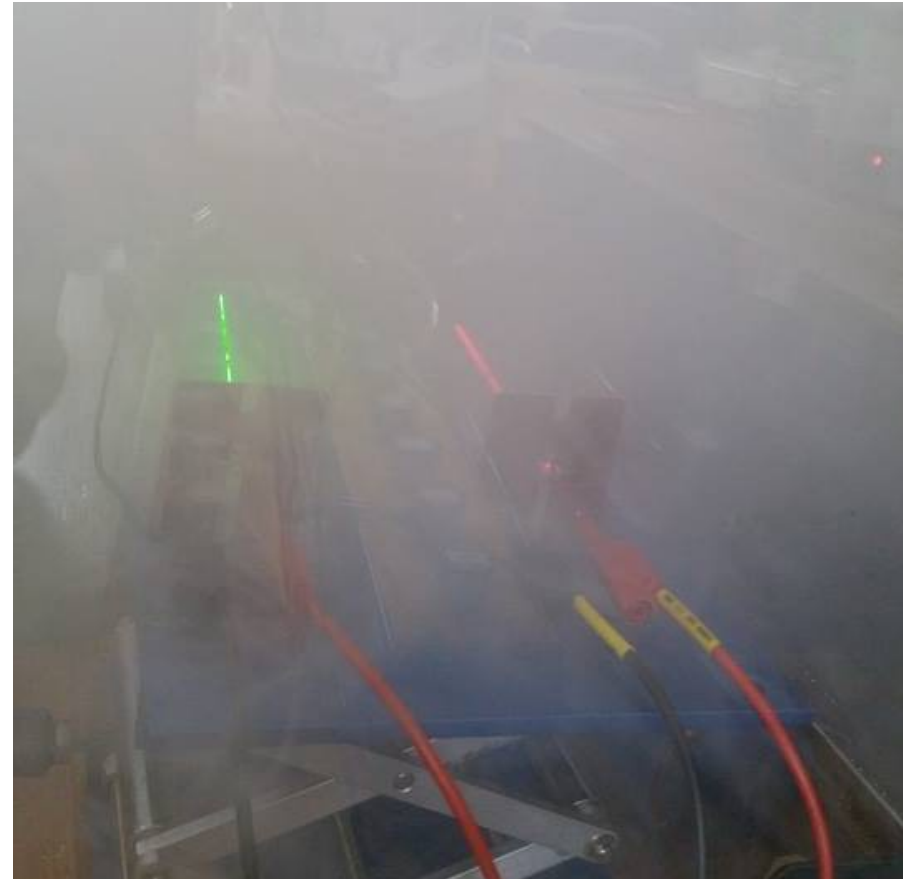
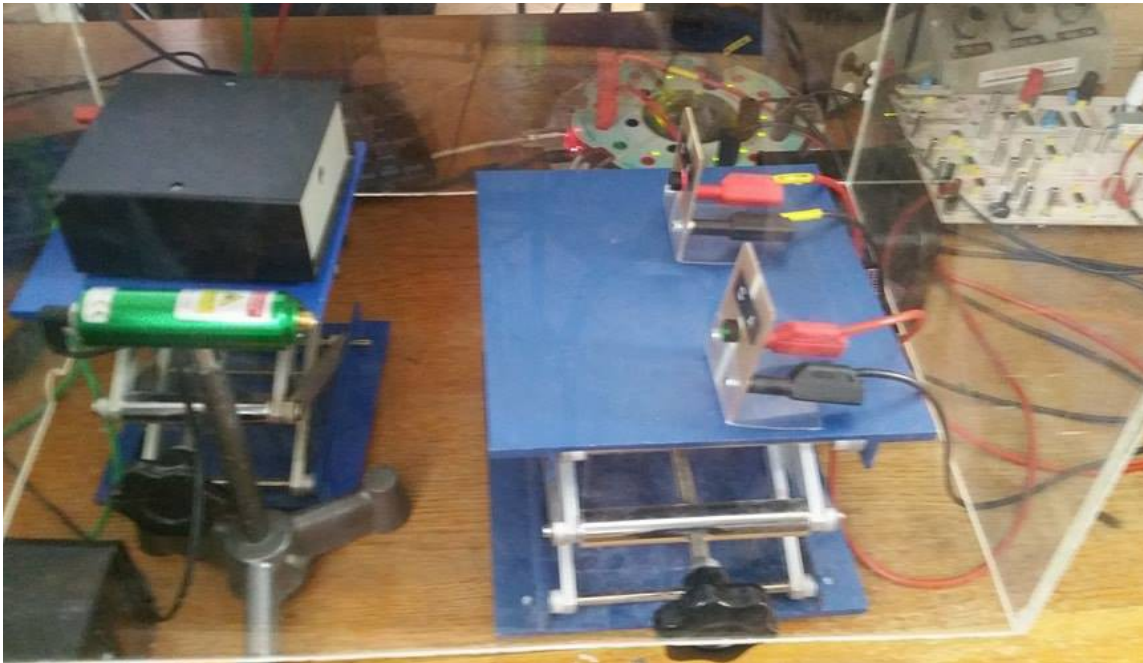
-Expérience :

- diode laser
- photodiode
- polariseur

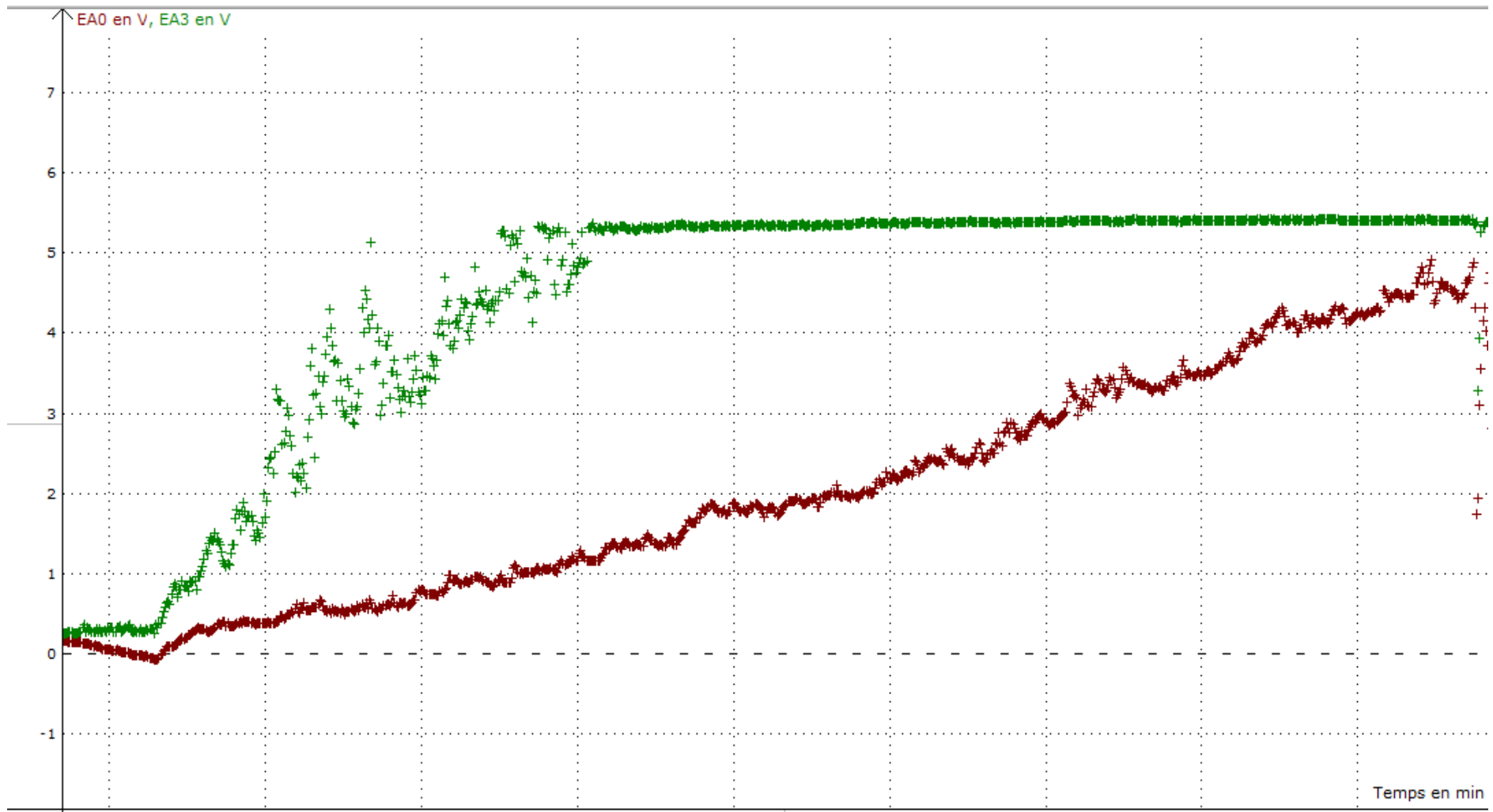
Résultat



Expérience



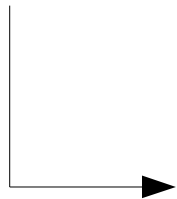
Résultats: tensions aux bornes des résistances



Analyse des résultats

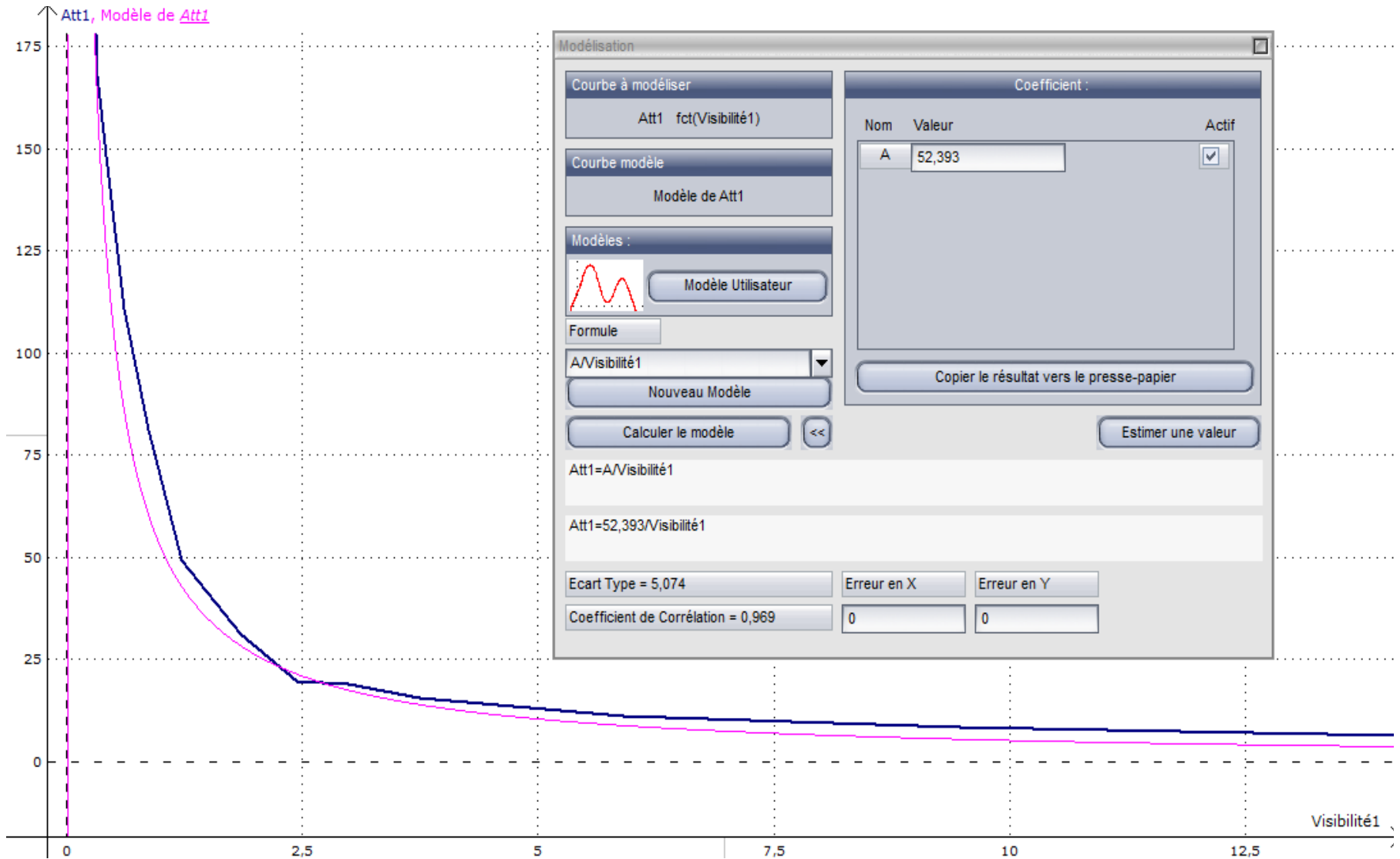
$$\begin{array}{l} \text{-Visibilité : } v = \frac{-\ln(0,05)}{\beta v} \\ \text{-Beer-Lambert : } \beta v = \frac{-\ln(T)}{L} \end{array} \left. \vphantom{\begin{array}{l} \text{-Visibilité : } v = \frac{-\ln(0,05)}{\beta v} \\ \text{-Beer-Lambert : } \beta v = \frac{-\ln(T)}{L} \end{array}} \right\} v = \frac{\ln(5) \times L}{\ln(T)}$$

$$\text{-Atténuation : } Att = \beta v \times \left(\frac{10}{\ln(10)} \right)$$

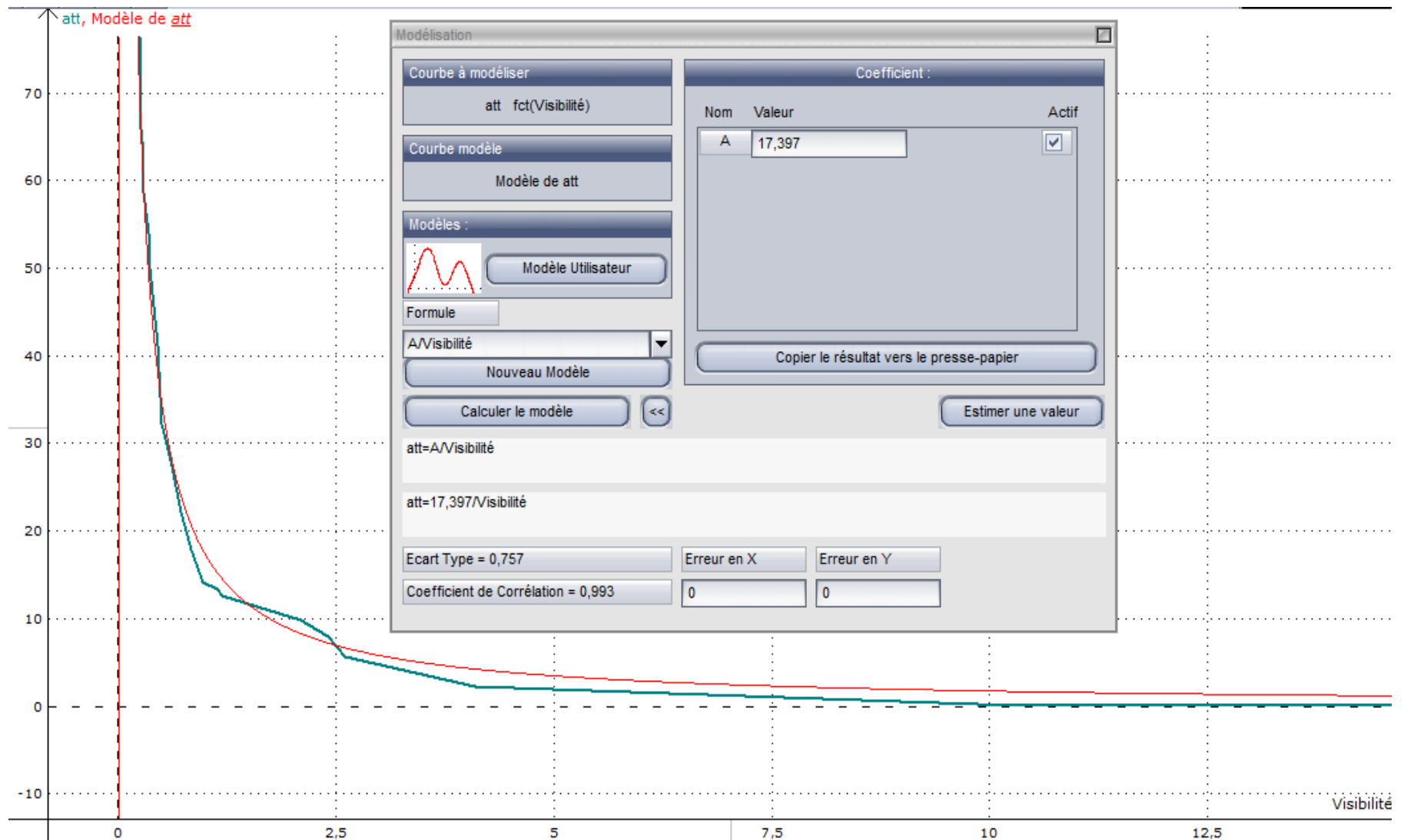


$$Att = \frac{10}{\ln(10)} \frac{-\ln(T)}{L}$$

$Att=f(V)$ pour $\lambda=480\text{nm}$

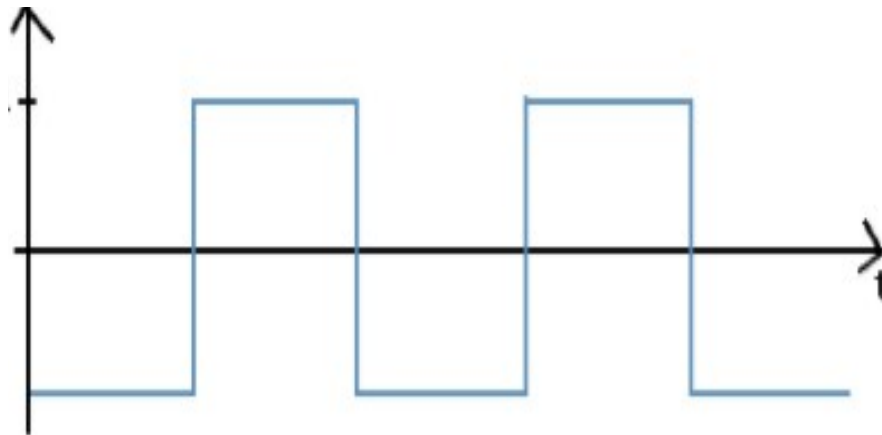


$Att=f(V)$ pour $\lambda=650\text{nm}$



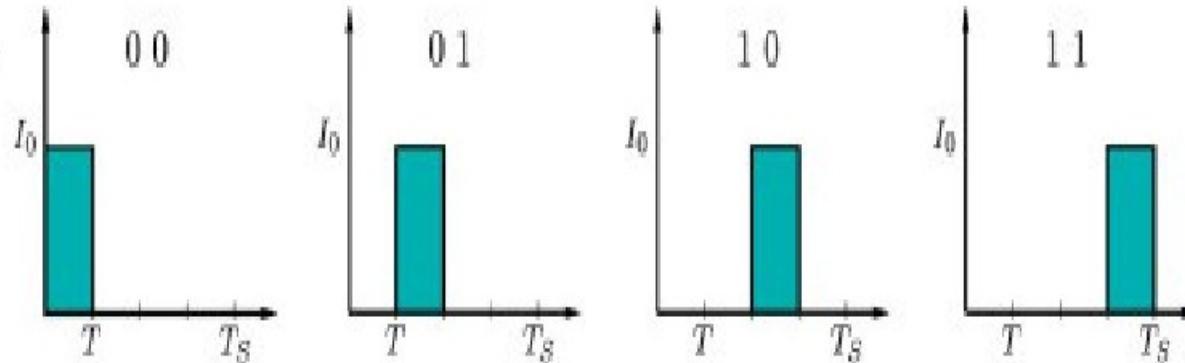
Modulations

- Modulation informatique
- Première modulation : OOK



- Débit binaire $D = \frac{1}{T_b}$

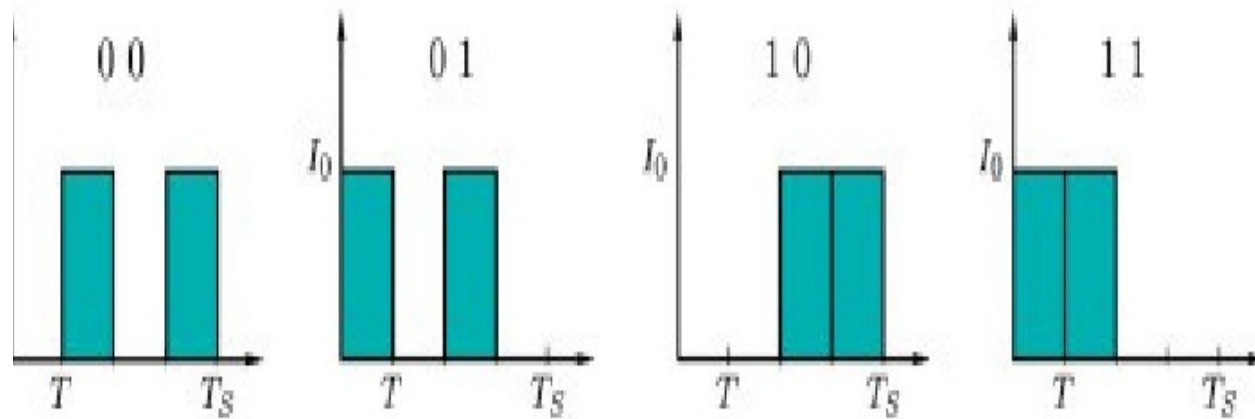
PPM



-Principe : b bits sera codé en M symboles de 2^b
intervalles de temps possible T_s
Chaque symbole est divisé en M sous intervalles
de temps T

$$T_s = \frac{T_b \log_2(M)}{M}$$

MPPM



-Principe : Un symbole divisé en M slots, l'émetteur envoie w impulsions chaque symbole

Nombre de symbole possible : $L = \binom{M}{w}$

Comparaison sur une série de 3bit

<i>OOK</i>	<i>PPM</i>	<i>MPPM</i>
000	00000001	00011
001	00000010	00101
010	00000100	01001
011	00001000	10001
100	00010000	00110
101	00100000	01010
110	01000000	10010
111	10000000	01100

Efficacité spectrale

-Définition : η est le rapport entre le débit binaire et la largeur de la bande passante

$$-\eta = \frac{D}{B} \text{ où } D = \frac{1}{T_b} \text{ et } B = \frac{1}{T}$$

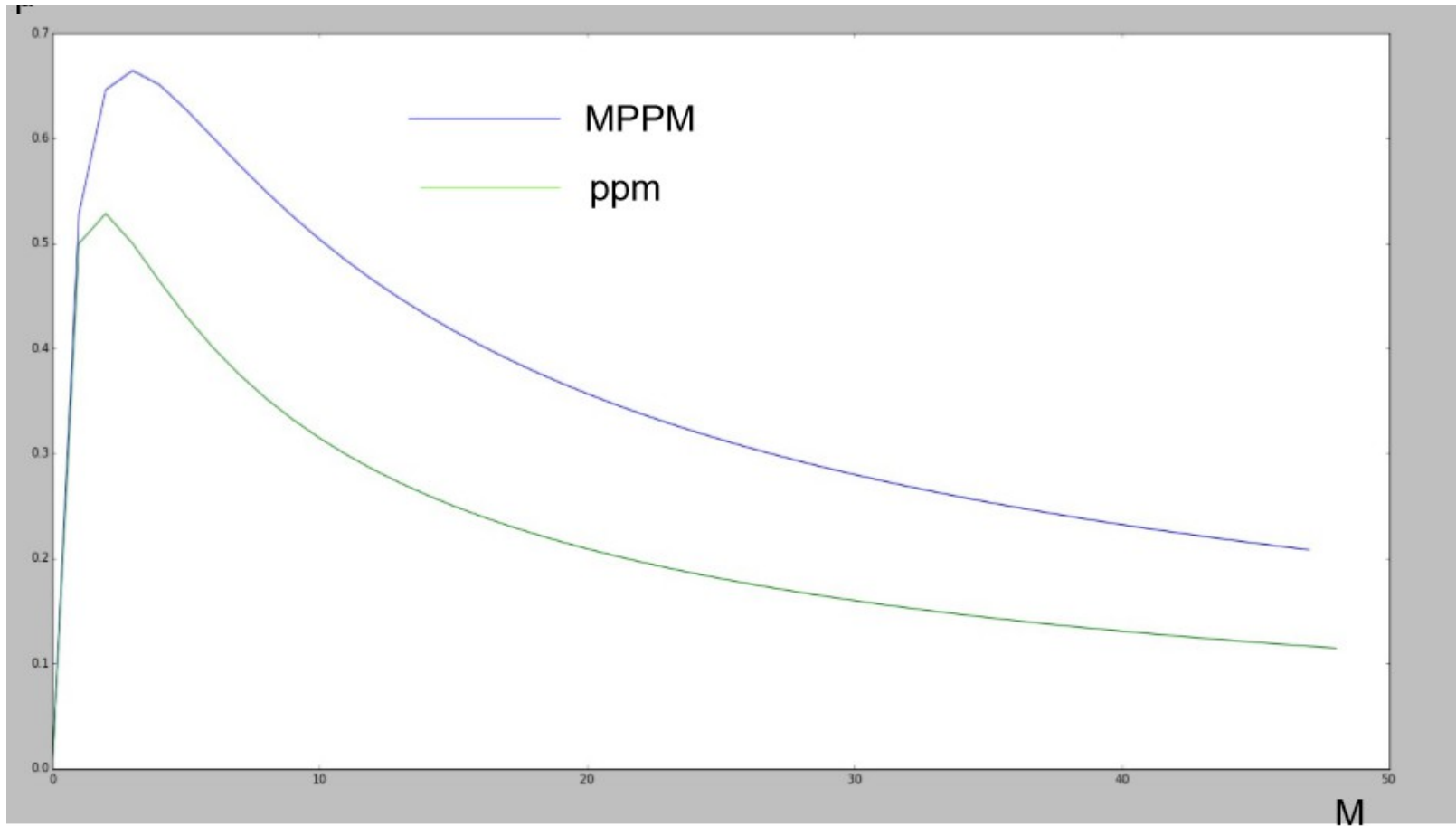
$$-\text{PPM} : \eta = \frac{\log_2(M)}{M}$$

$$-\text{OOK} : \eta = 1 \text{ bit/s/Hz}$$

$$-\text{MPPM} : \eta = \frac{\log_2(L)}{M}$$

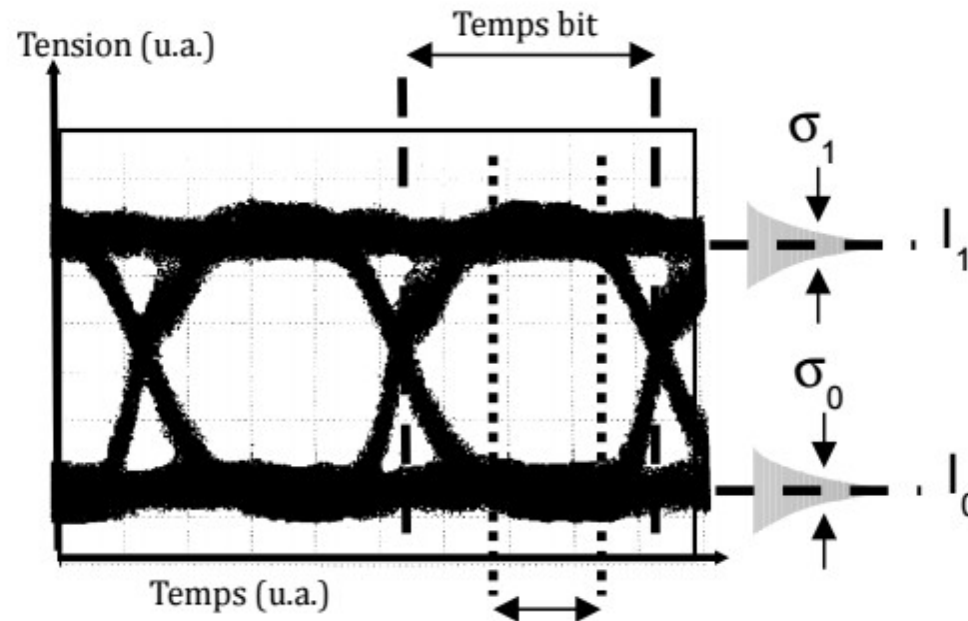
$$\eta = f(M) \text{ pour } w=2$$

$\eta(\text{bit/s/Hz})$



Étude expérimentale

- Modulation informatique de la diode
- Diagramme de l'œil :



- Facteur de qualité : $Q = \frac{I_1 - I_0}{\sigma_1 + \sigma_0}$

Diagramme de l'œil OOK

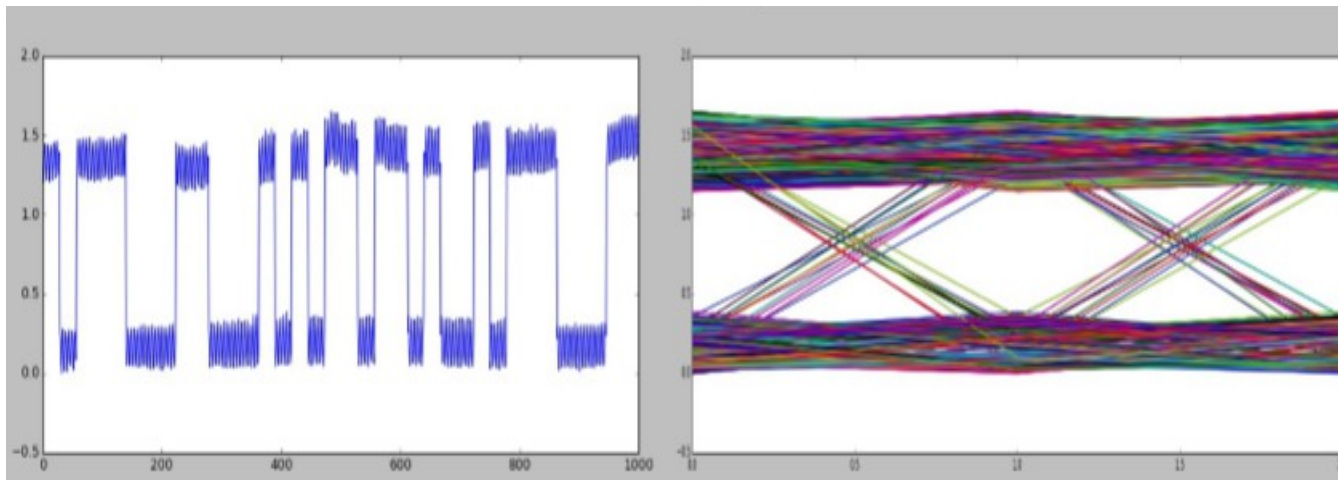
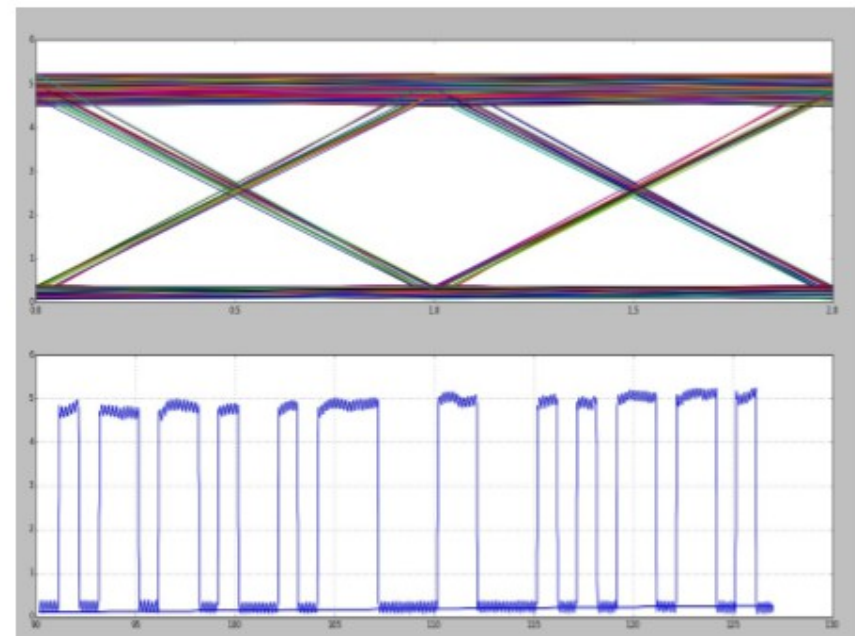
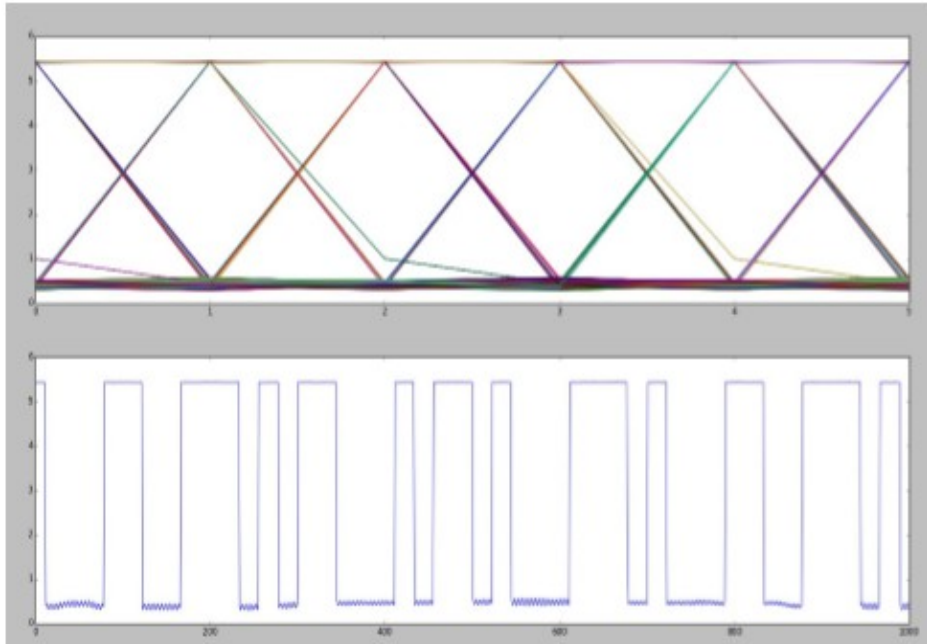


Diagramme de l'œil PPM

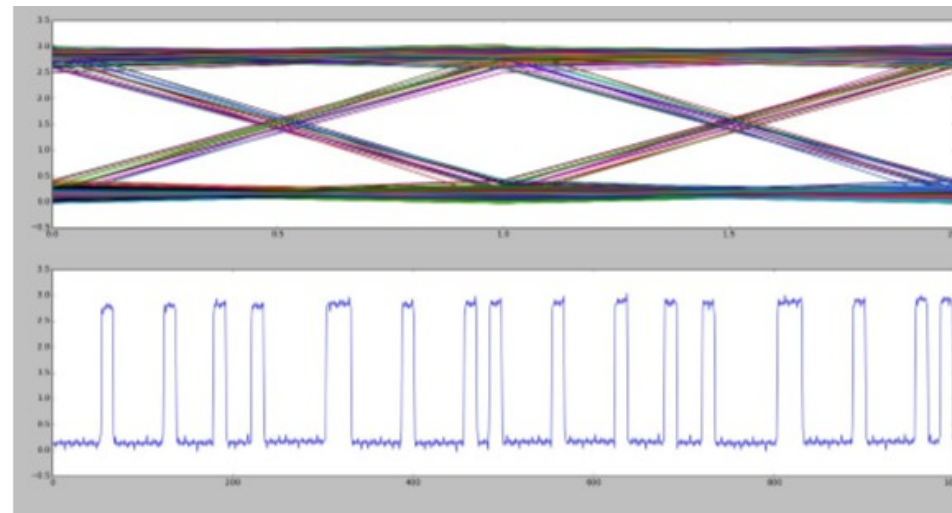
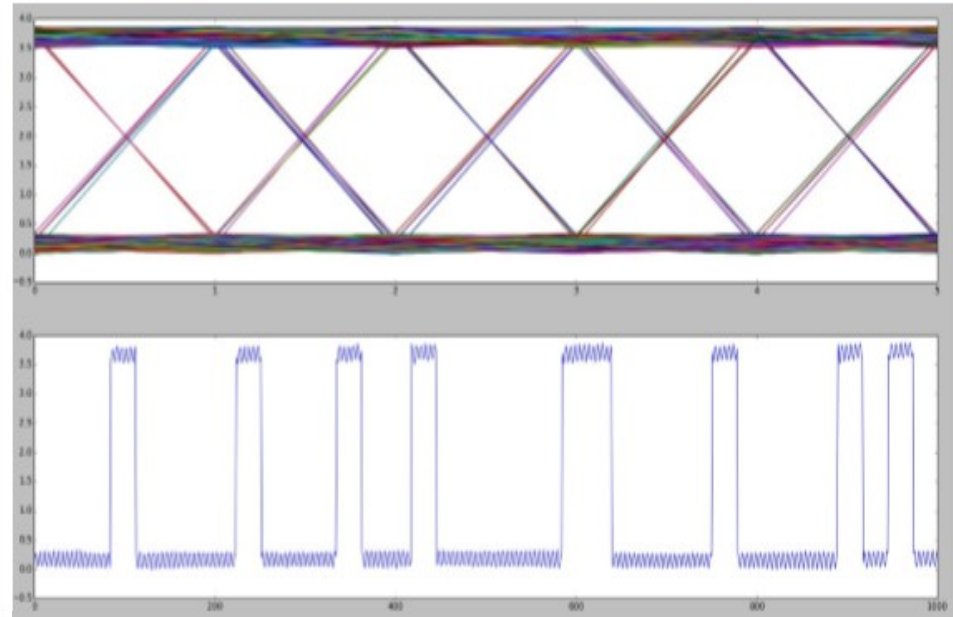
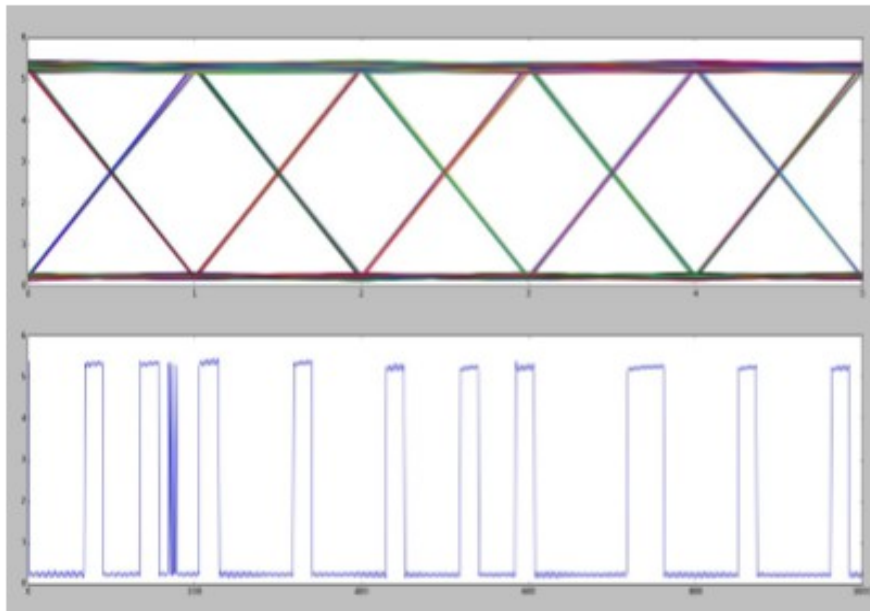


Diagramme de l'œil PPM

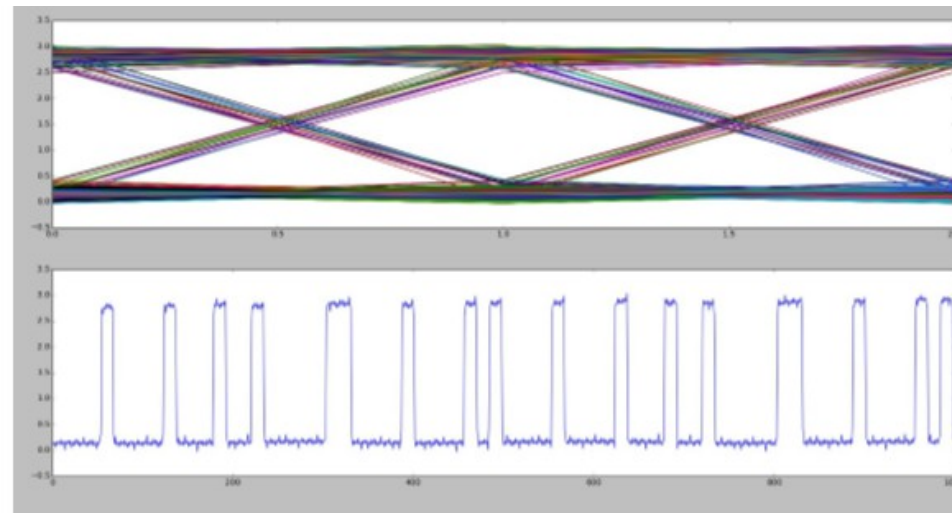
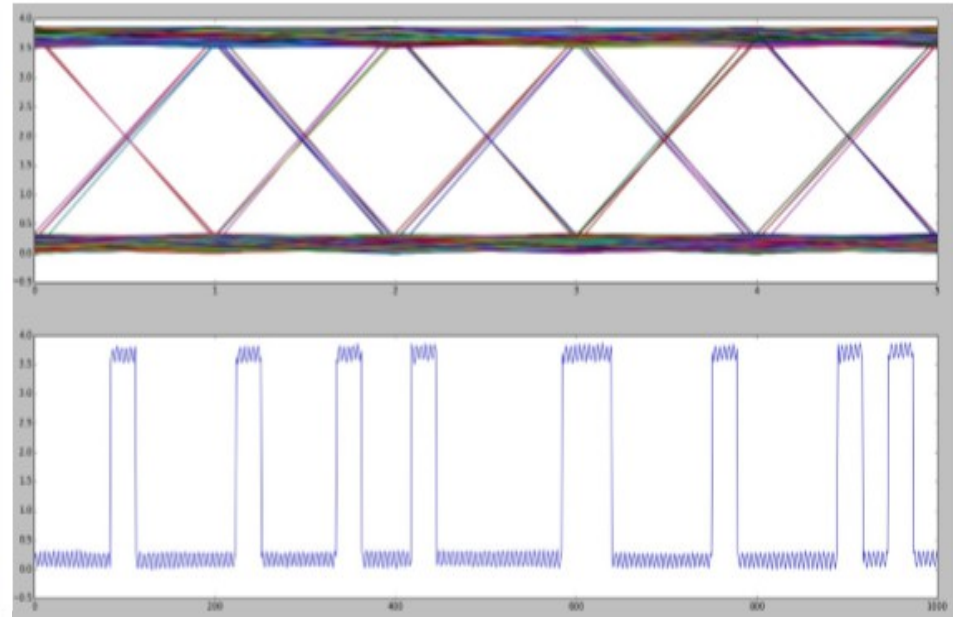
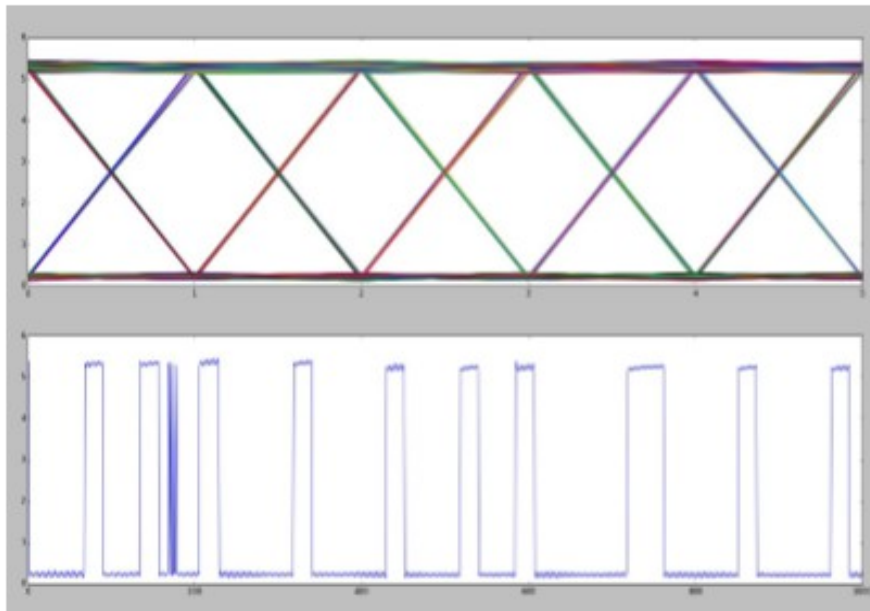
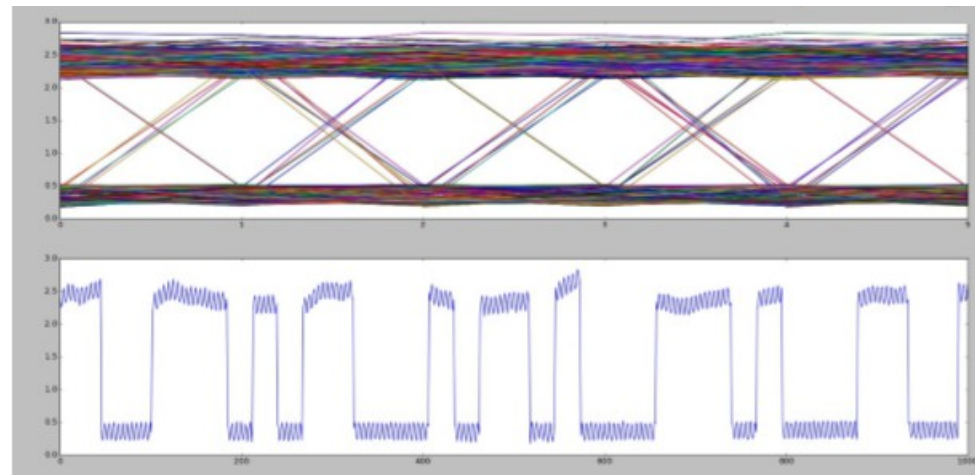
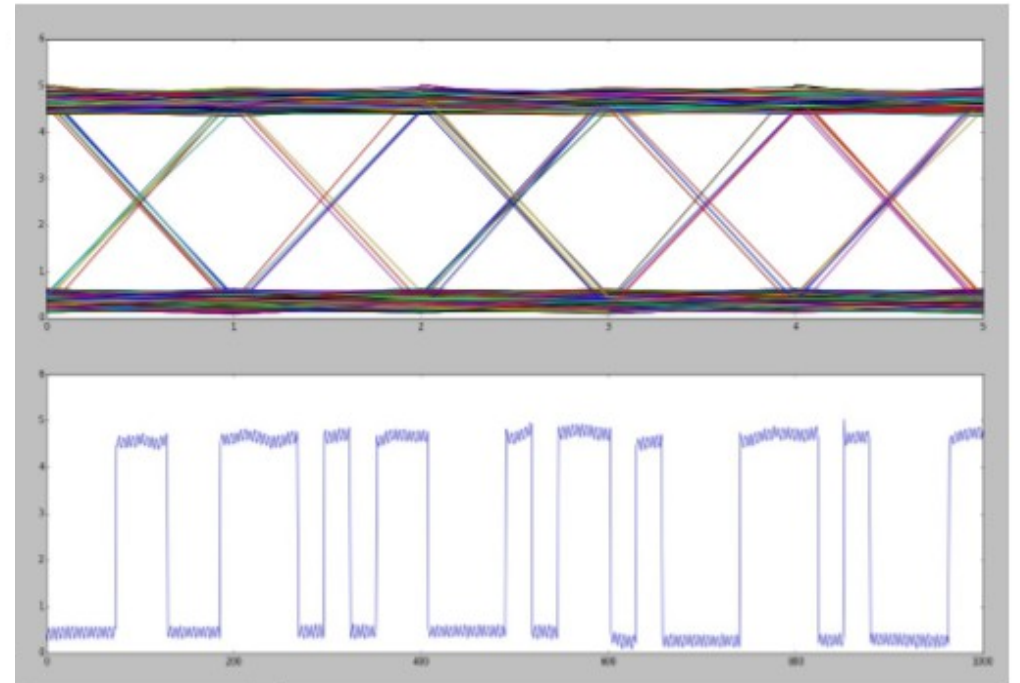
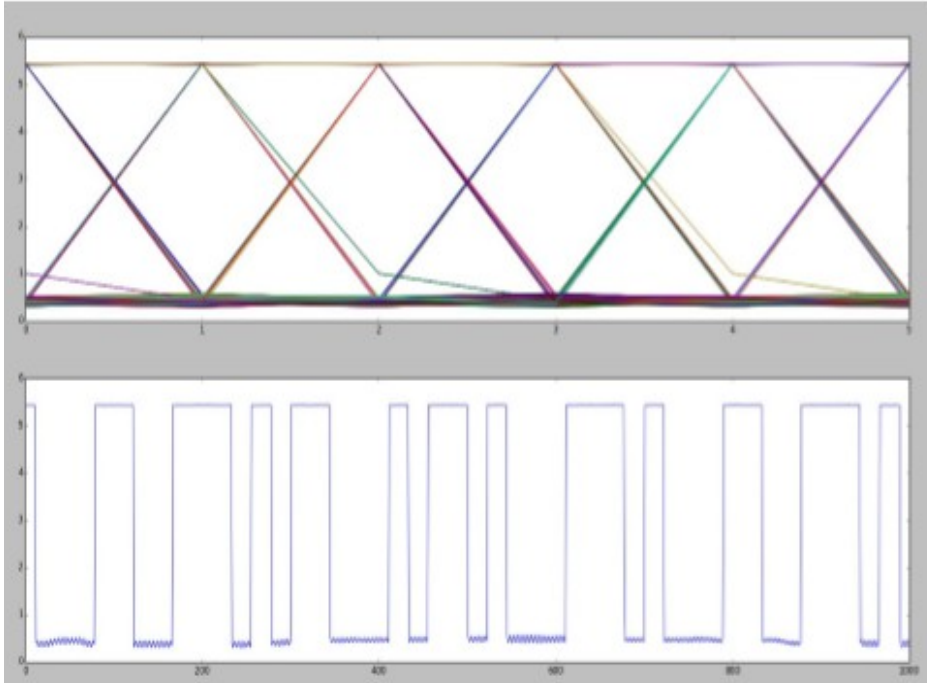
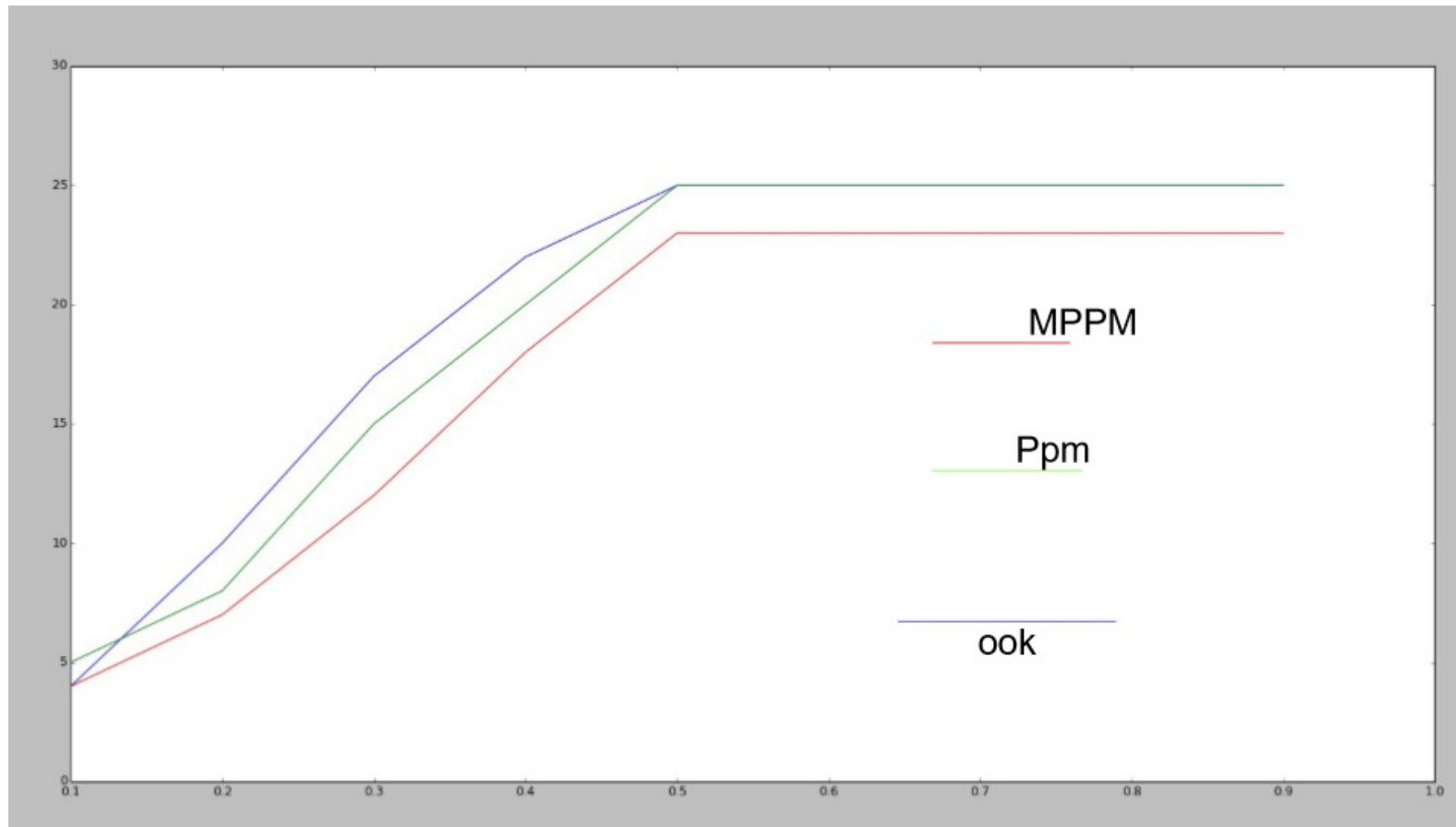


Diagramme de l'œil MPPM



$$Q=f(T)$$

Q



T

Conclusion

-Pour la longueur d'onde :

LOA → utilisé une longueur d'onde importante

$$Att \propto \frac{\lambda^{-q}}{V}$$

-Pour les types de modulations :

Efficacité spectrale → MPPM

Facteur de qualité → OOK



modulations hybrides

Annexe

Résultat Malus

U(V)	$\theta(^{\circ})$	1,220	45
0,083	0	1,425	50
0,069	5	1,645	55
0,093	10	1,89	60
0,163	15	2,141	65
0,259	20	2,305	70
0,405	25	2,603	75
0,565	30	2,622	80
0,770	35	2,709	85
1	40	2,800	90

Valeurs 480nm

Atténuation(dB/m)	Visibilité(m)		
273,176	0,199	18,941	2,984
208,885	0,289	15,534	3,747
167,874	0,33	11,278	5,897
148,2	0,421	8,547	9,471
110,7	0,61	3,789	20,926
81,241	0,87	1,068	31,478
49,317	1,214		
31,148	1,847		
19,57	2,441		

Valeurs 650nm

Atténuation(dB/m)	Visibilité(m)		
100	0,16	17,8	0,84
77	0,25	14,2	0,97
65,9	0,27	13,4	1,14
63	0,296	12,6	1,2
58,7	0,3	9,8	2,1
54	0,348	7,9	2,41
50	0,361	5,7	2,6
41,2	0,45	2,2	4,1
32,2	0,50	0,14	10,3
28	0,61	0,009	16,8
22,2	0,72		

Arduino OOK

```
String myString = "101110001101";

void setup() {
  Serial.begin(115200);
  pinMode(13, OUTPUT);
}

void loop() {
  for(int i=0;i<26;i++){
    if(myString[i]=='1'){
      digitalWrite(13,HIGH);
      delay(1000);
    }
    else if(myString[i]=='0'){
      digitalWrite(13,LOW);
      delay(1000);
    }
  }
}
```

Arduino PPM

```
int i = 0;
String mystring = "001011100000111010101111000110111
void setup() {
  Serial.begin(115200);
  pinMode(13,OUTPUT);
}

void loop() {
  while(i<=124){
    if(mystring[i]== "0" && mystring[i+1]=="0"){
      digitalWrite(13,HIGH);
      delay(1000);
      digitalWrite(13,LOW);
      delay(3000);
    }
    else if (mystring[i]== "0" && mystring[i+1]=="1"){
      digitalWrite(13,LOW);
      delay(1000);
      digitalWrite(13,HIGH);
      delay(1000);
      digitalWrite(13,LOW);
      delay(2000);
    }
    else if (mystring[i]== "1" && mystring[i+1]=="0"){
      digitalWrite(13,LOW);
      delay(2000);
      digitalWrite(13,HIGH);
      delay(1000);
      digitalWrite(13,LOW);
      delay(1000);
    }
    else if (mystring[i]== "1" && mystring[i+1]=="1"){
      digitalWrite(13,LOW);
      delay(3000);
      digitalWrite(13,HIGH);
      delay(1000);
    }
    i=i+2;
  }
}
```

Arduino MPPM

```
String mystring = "001011100000111010101111000110111111  
void setup() {  
  Serial.begin(115200);  
  pinMode(13,OUTPUT);  
  
}  
  
void loop() {  
  while(i<=124){  
    if(mystring[i]== "0" && mystring[i+1]=="1"){  
      digitalWrite(13,HIGH);  
      delay(1000);  
      digitalWrite(13,LOW);  
      delay(1000);  
      digitalWrite(13,HIGH);  
      delay(1000);  
      digitalWrite(13,LOW);  
      delay(1000);  
    }  
    else if (mystring[i]== "1" && mystring[i+1]=="0"){  
      digitalWrite(13,LOW);  
      delay(2000);  
      digitalWrite(13,HIGH);  
      delay(2000);  
    }  
    i=i+2;  
  }  
  else if (mystring[i]== "1" && mystring[i+1]=="1"){  
    digitalWrite(13,HIGH);  
    delay(2000);  
    digitalWrite(13,LOW);  
    delay(2000);  
  }  
  else if (mystring[i]== "0" && mystring[i+1]=="0"){  
    digitalWrite(13,LOW);  
    delay(1000);  
    digitalWrite(13,HIGH);  
    delay(1000);  
    digitalWrite(13,LOW);  
    delay(1000);  
    digitalWrite(13,HIGH);  
    delay(1000);  
  }  
}
```

Diagramme de l'œil python

```
from pylab import *
signal = '0,4672651738 0,4424106306 0,3579051839 0,3877306357 0,4622942652 0,437439722 0,3628760925
temps = '0,72 0,765 0,81 0,855 0,9 0,945 0,99 1,035 1,08 1,125 1,17 1,215 1,26 1,305

def transforme(s):#transforme une chaine de caractère en liste de nombres
    s = s.replace(',','.')
    s = s.split()
    for k in range (len(s)):
        s[k]=float(s[k])

    return(s)

subplot(211)#trace le diagramme de l'oeuil
u = 0
n = len(transforme(signal))
while u < n :
    plt.plot(transforme(signal)[u:u+3])
    u = u+3

subplot(212)#trace le signal reçu en fonction du temps
plt.grid()
plt.plot(transforme(temps),transforme(temps))
plt.show()
```