



Epreuve TIPE Session 2019

Transport de l'information par LASER

49554



Plan

- ☐ Introduction
- ☐ Importance du laser
- ☐ Problématique & Objectifs
- ☐ Transport de l'information par laser
- ☐ Modélisation informatique
- ☐ Applications
- ☐ Conclusion et perspectives

Introduction

Pigeon voyageur



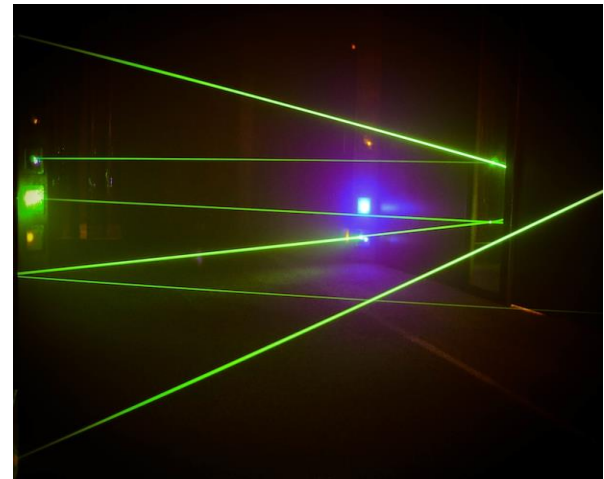
Communication par télégraphe



Communication par satellite



Communication par laser



An abstract background on the left side of the slide featuring vibrant green laser light beams crisscrossing against a black backdrop, with several bright green circular bokeh effects scattered throughout.

Importance du laser

Propriétés du laser

- **Quasi-monochromaticité**
- **Puissance**
- **directivité**

Importance du laser

- **Faible atténuation au cours de la propagation**
- **Transmission par fibre optique**
- **Communication à haut débit et à grande capacité**
- **Un large domaine d'applications:
spectroscopie, industrie, scanner, imprimantes,
...)**



Problématique et objectifs

Problématique:

La transmission d'une information sonore à l'air libre présente plusieurs inconvénients.

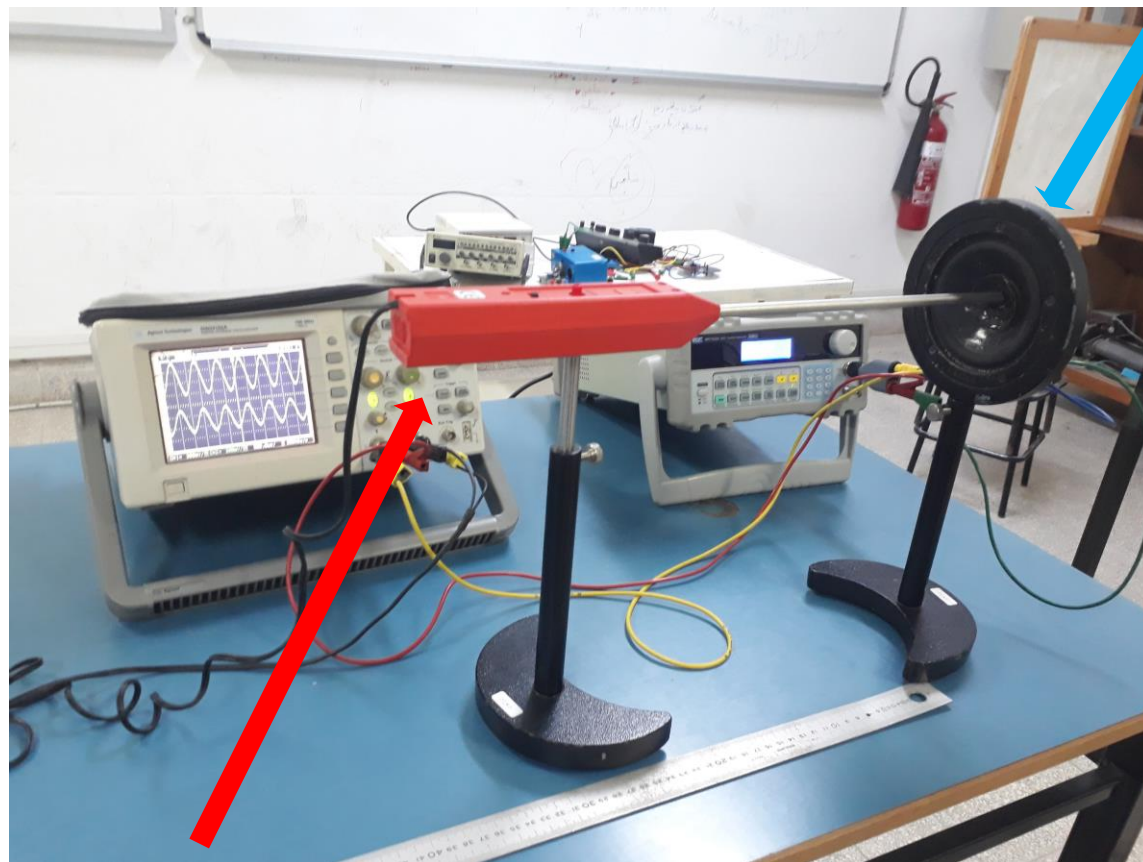
- **Comment peut-on optimiser ce phénomène de transport au moyen d'un laser?**
- **Quels sont les limites de cette technique et quelles sont les améliorations possibles?**
- **Quelle applications peut-on en déduire?**

Objectifs

- **Détermination du diagramme de rayonnement d'un haut parleur**
- **Mise en evidence expérimentale de l'intérêt d'utilisation d'un laser pour transmettre du son**
- **Critiques de cette méthode**
- **Présenter quelques applications**

Etude de la propagation du son dans l'air libre

I) Montage expérimental



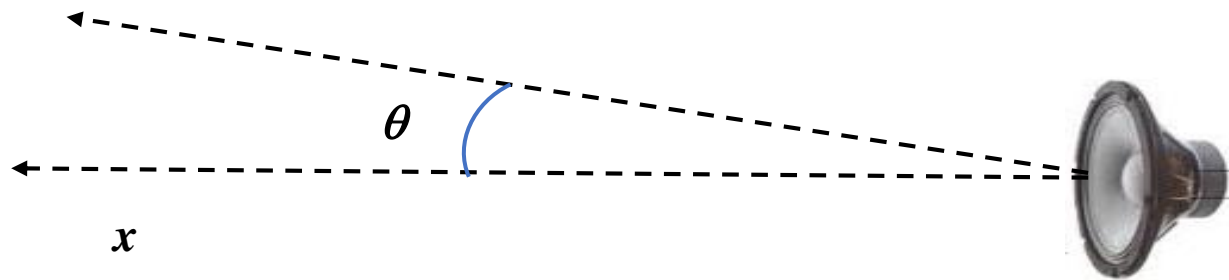
Haut parleur

Microphone

Inconvénients de la propagation libre dans l'air

I-Effet de la distance x

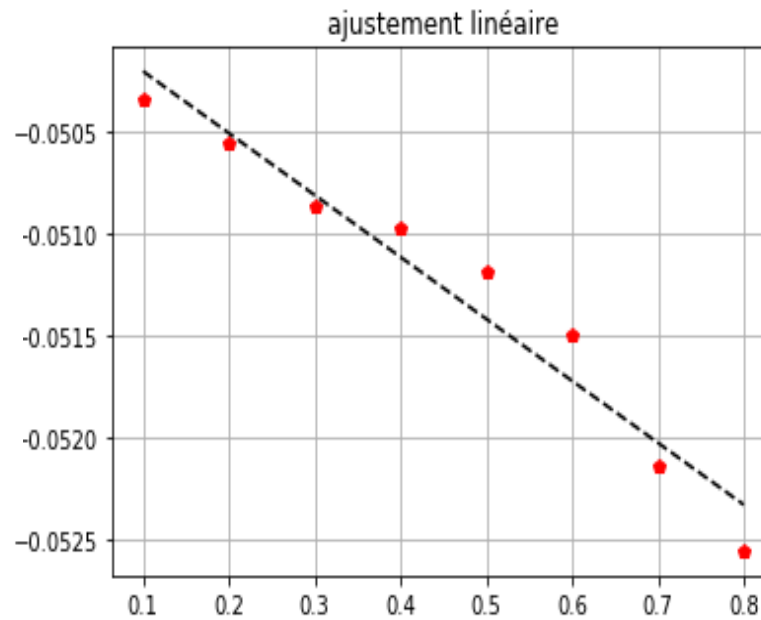
$$\theta = 0$$



$x(m)$	0,1	0,2	0,3	0,4
$-\ln\left(\frac{A(x)}{A(x=0)}\right) * 10^2$	5,0346	5,0557	5,0872	5,0978
$x(m)$	0,5	0,6	0,7	0,8
$-\ln\left(\frac{A(x)}{A(x=0)}\right) * 10^2$	5,1188	5,1504	5,2136	5,2557



$$\ln\left(\frac{A(x)}{A(x=0)}\right)$$



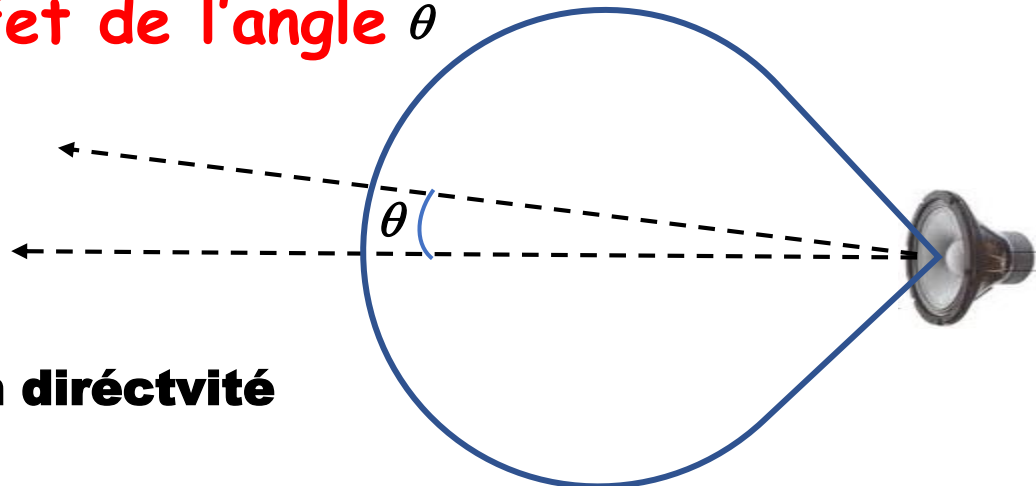
$$A(x) = A(x = 0)e^{-k''x}$$

$$x(m)$$

$$\text{Pente } k'' = 3 \cdot 10^{-3} m^{-1}$$

- Atténuation**

II-Effet de l'angle θ

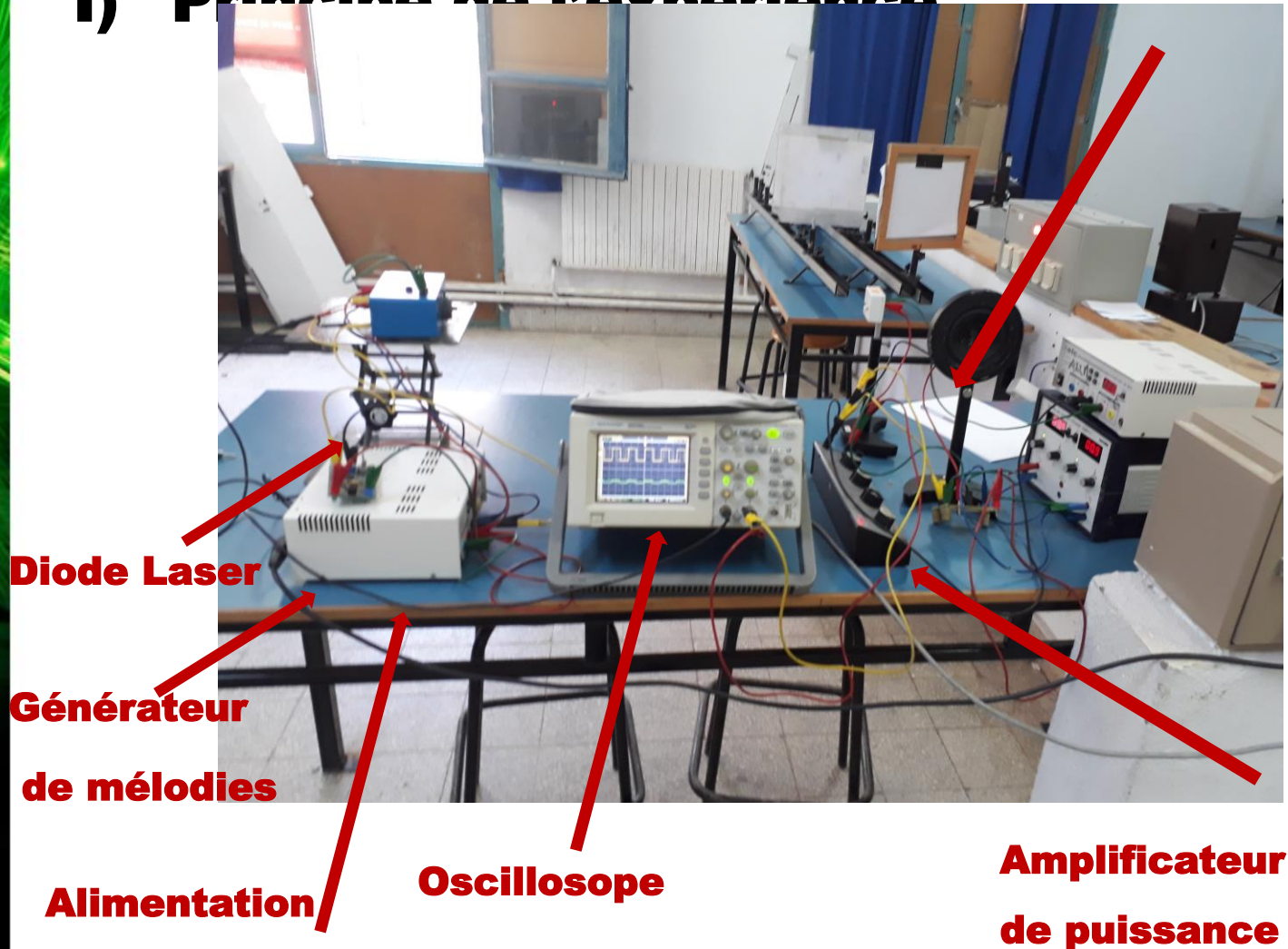


- Non directivité**

Transport de l'information sonore par laser

I) Principe de l'expérience

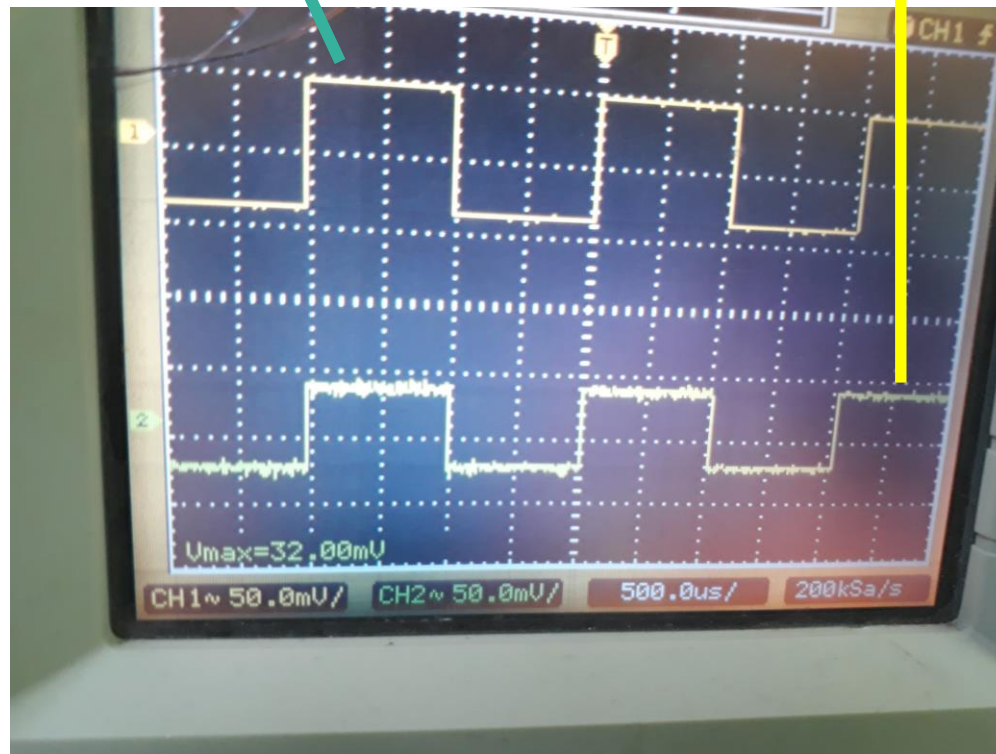
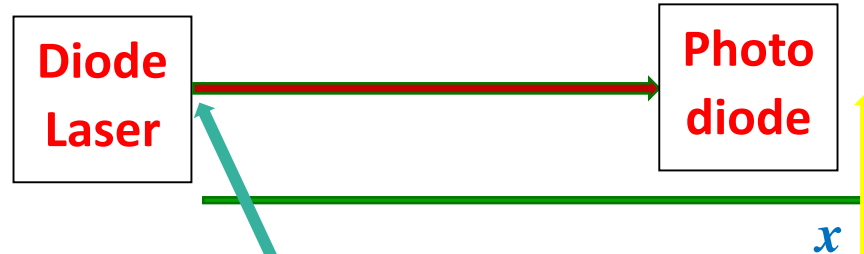
Haut parleur



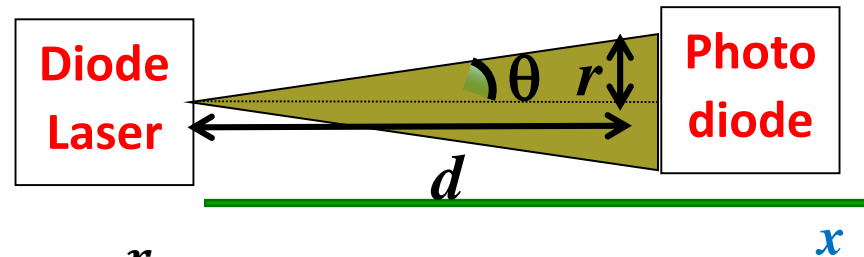
II) Mise en evidence de la modulation



III) Etude de la propagation dans l'air



II) Elargissement, perte énergétique



$$\tan \theta \sim \theta = \frac{r}{d}$$

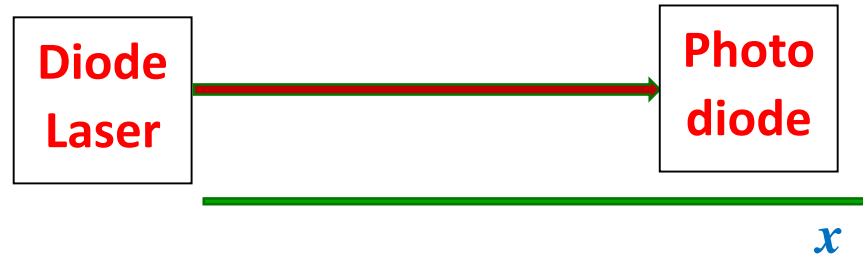
X(cm)	10	20	30	40	50	60	70	80
θ (rad)	0,005	0,005	0,005	0,005	0,005	0,005	0,005	0,005

surface de la photodiode : $s = 16\text{mm}^2$

X(cm)	10	20	30	40	50	60	70	80
$\frac{P}{P_1}$	1	0,795	0,662	0,530	0,397	0,265	0,182	0,101
$\frac{S_1}{S}$	1	0,774	0,64	0,53	0,395	0,251	0,155	0,098

II) Utilité de la fibre optique

On considère une distance $d = 5\text{m}$

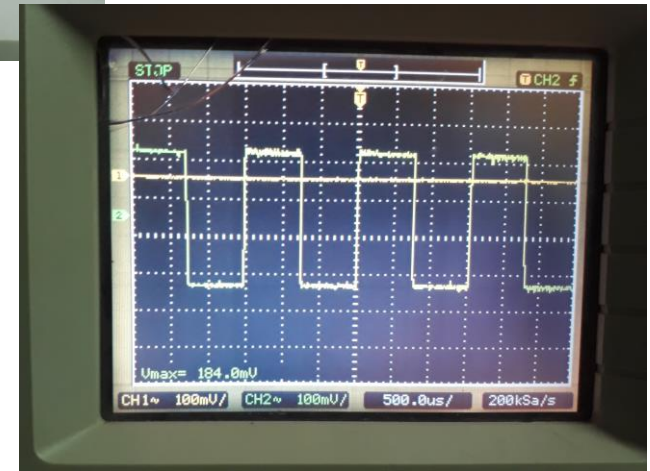


Air libre



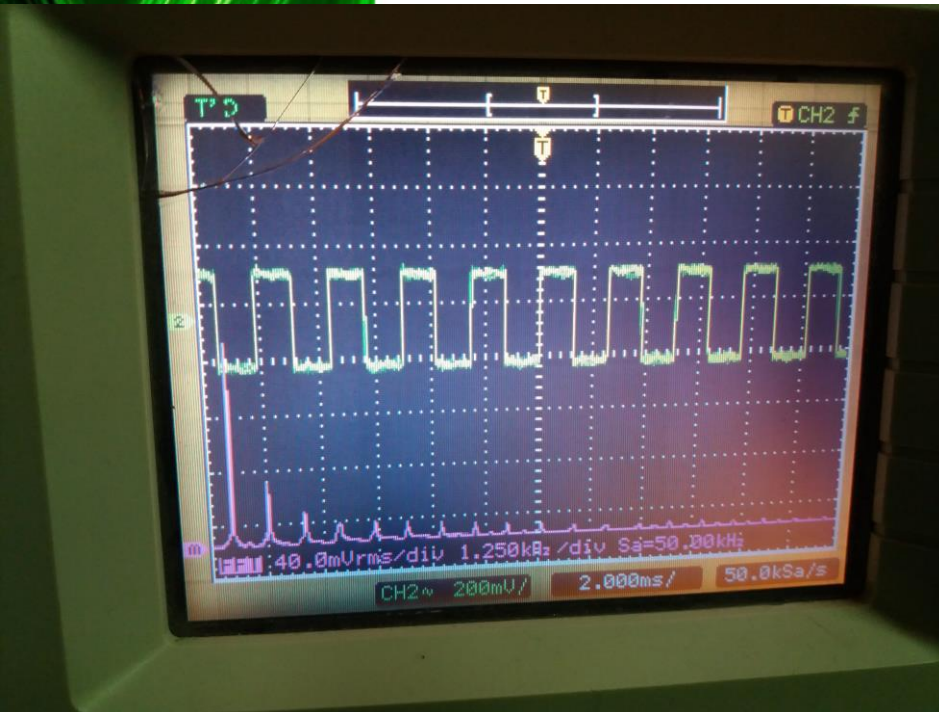
Avec la fibre optique

$$\frac{A(fibre)}{A(air)} = 9,3$$

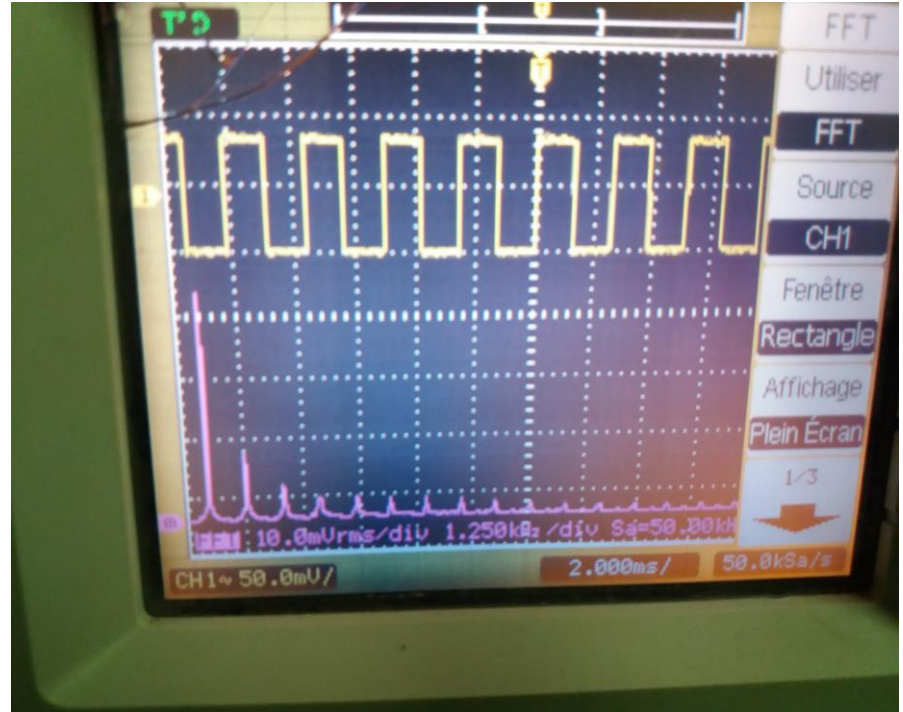


II) Effet de la photodiode

$f = 400\text{Hz}$, (Amplification = 5)



entrée



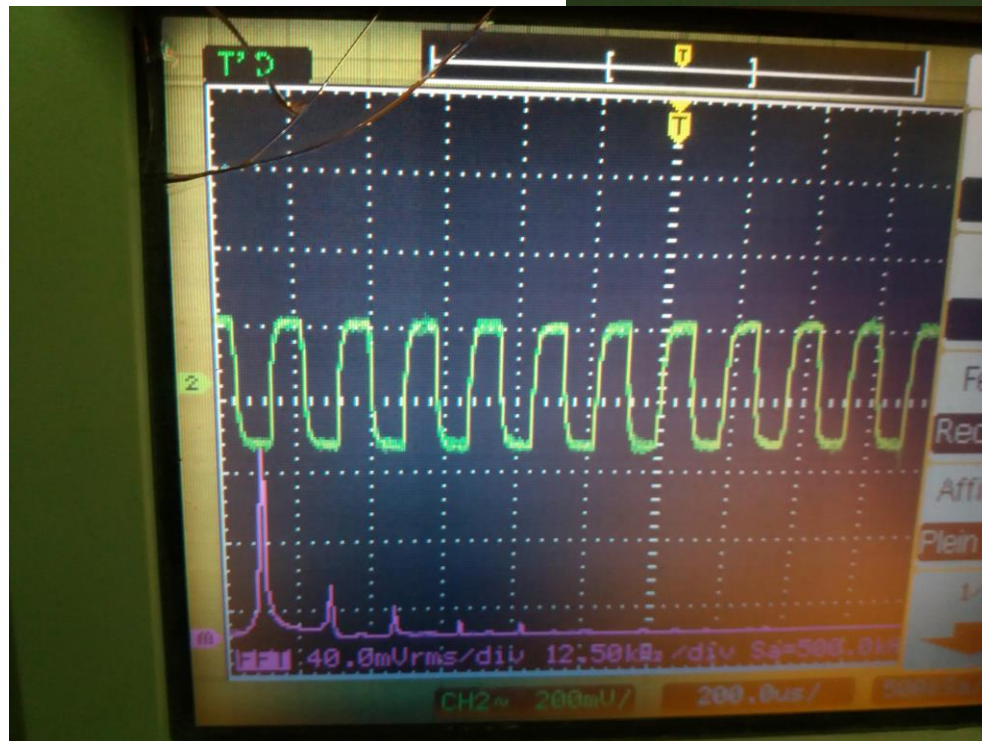
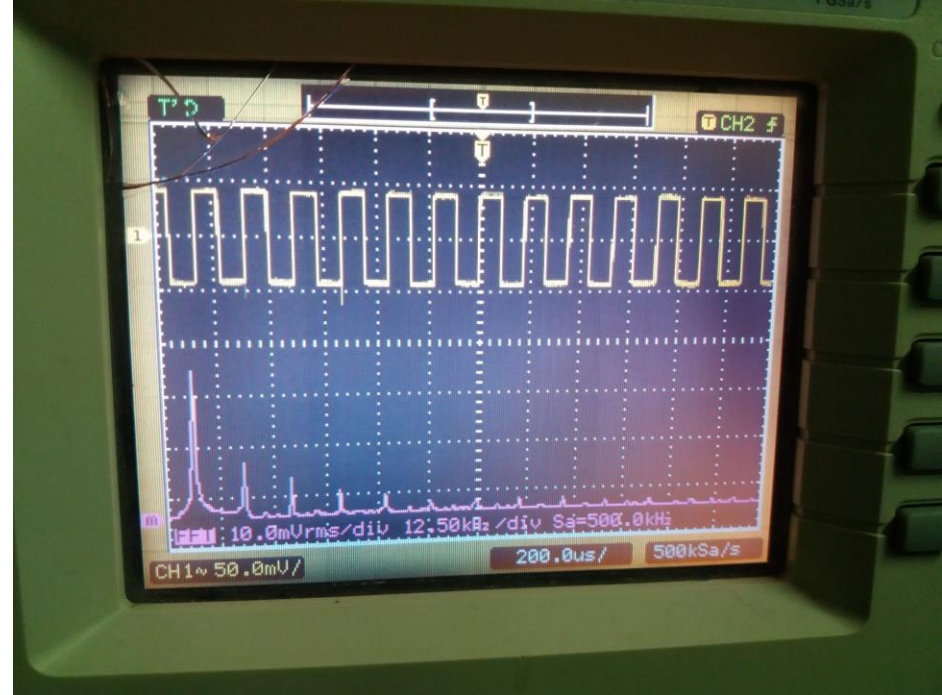
Sortie

Le spectre est conservé en amplitude et en fréquence: Le signal est reproduit fidèlement.



f = 5kHz

entrée

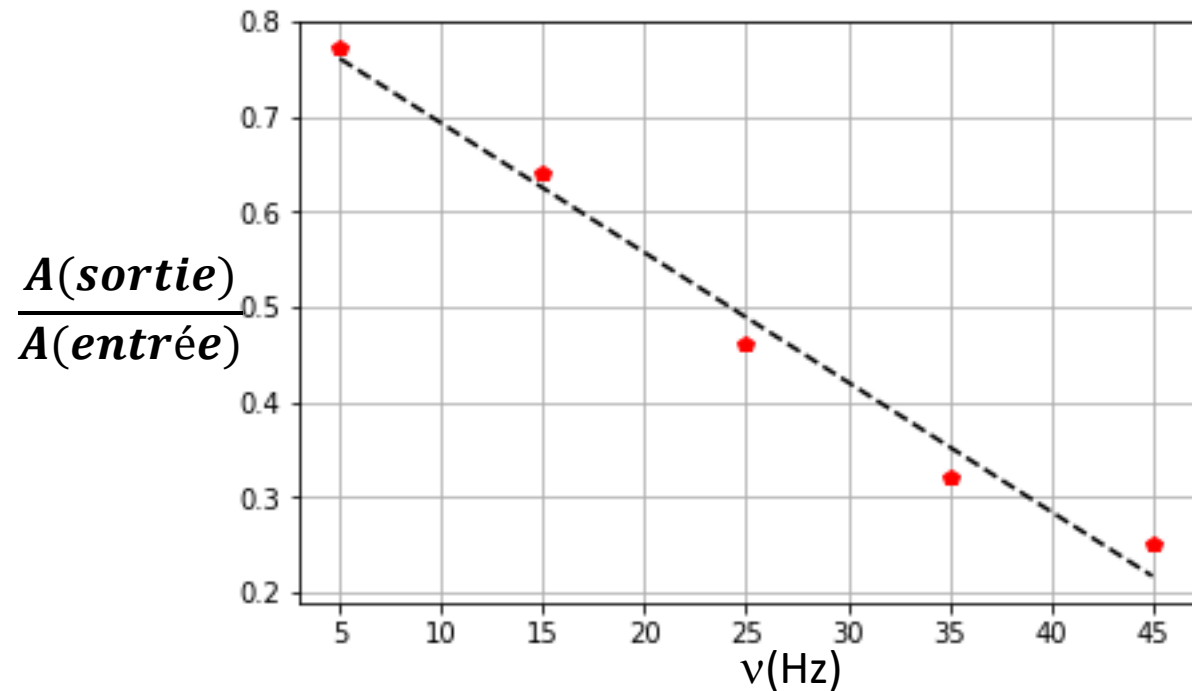


Sortie

f = 5kHz

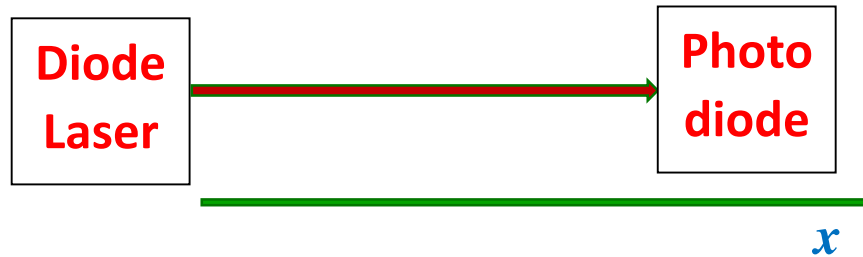
ν (kHz)	5	15	25	35	45
Aentrée(mV)	28	10	8	5	3,8
Asortie(mV)	21,6	6,4	3,7	1,6	0,96

Temps de réponse: de la photodiode: 0,1 μ s



II) Propagation dans la fibre optique

On considère une distance $d = 5\text{m}$



Equation de propagation, relation de dispersion

$$\Delta \vec{E} - \frac{n^2}{c^2} \frac{\partial^2 \vec{E}}{\partial t^2} = \vec{0}$$

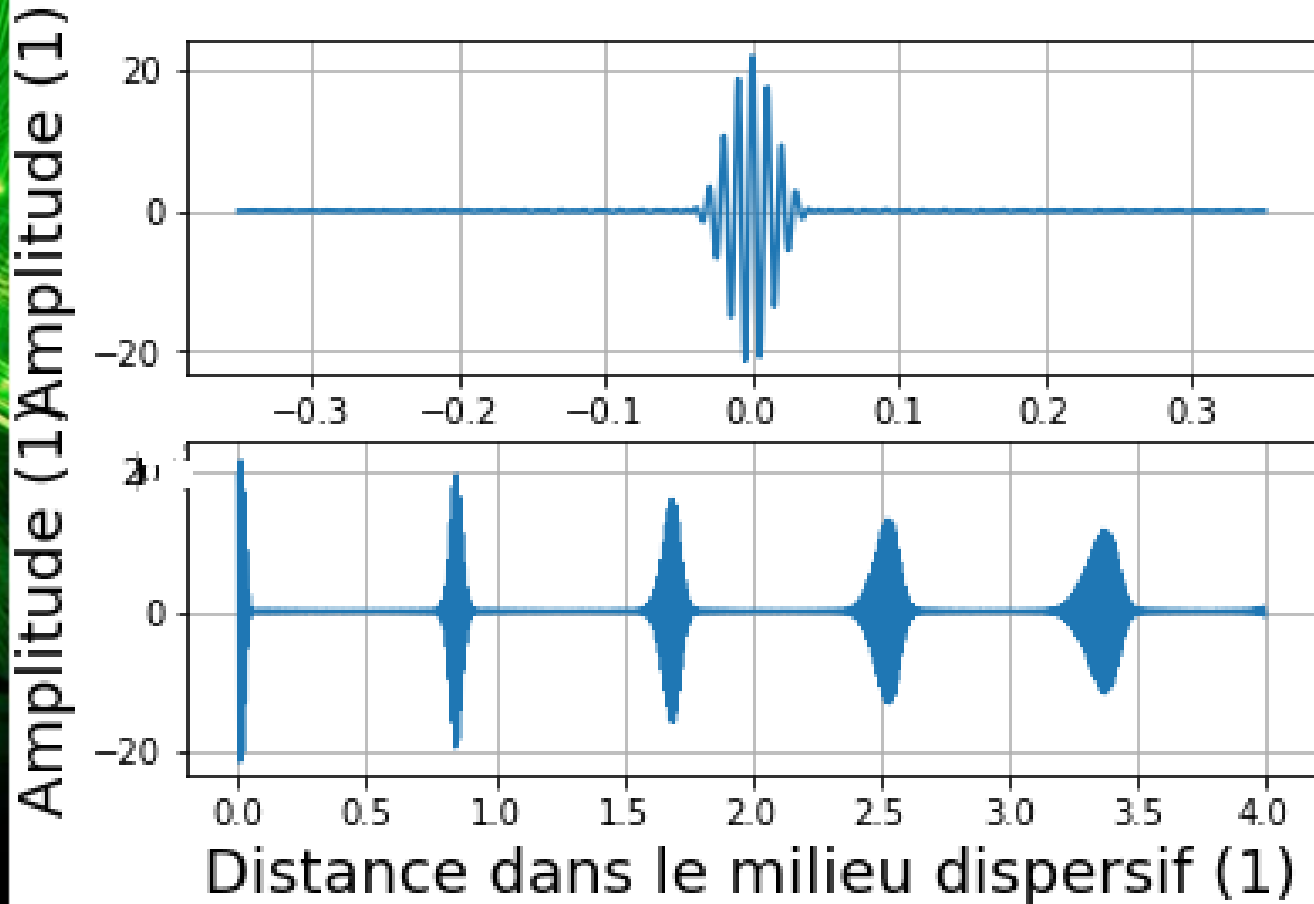
Paquet d'ondes

$$E = f(r, \theta) e^{i(\omega_0 t - k_0 z)}$$

Relation de dispersion dans la fibre

$$k(\omega) = k_0 + \left(\frac{\partial k}{\partial \omega} \right)_{\omega_0} (\omega - \omega_0) + \frac{1}{2} \left(\frac{\partial^2 k}{\partial \omega^2} \right)_{\omega_0} (\omega - \omega_0)^2$$

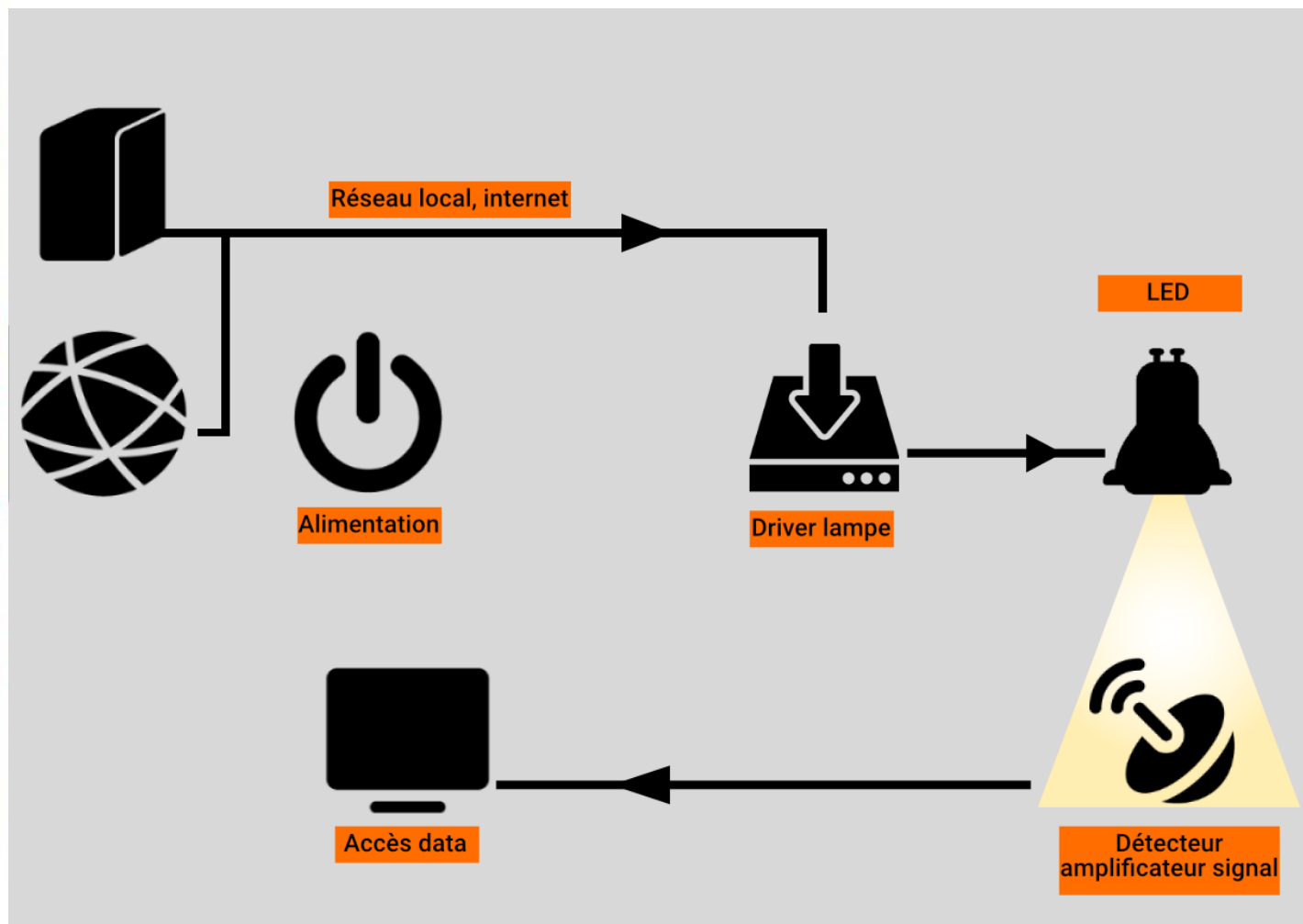
Simulation de l'effet de la fibre sur le transport d'information



Applications

LIFI: Une nouvelle technologie





Finance des marchés



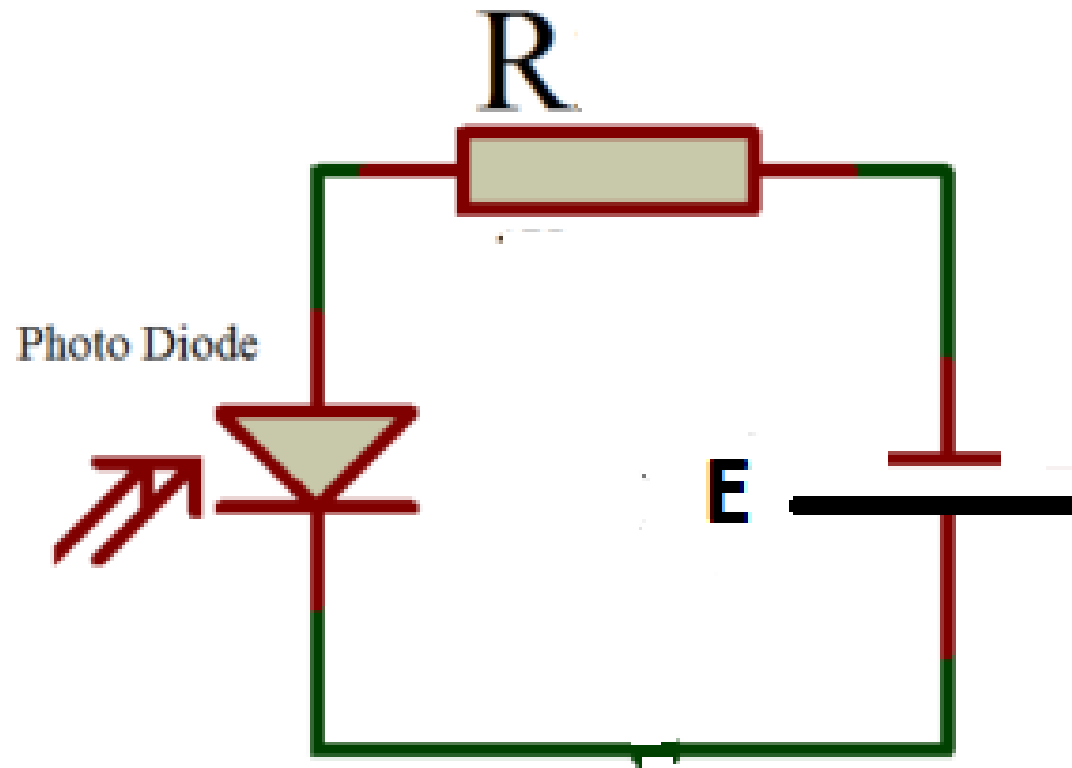
Trading hautes fréquences



Critiques et conclusion

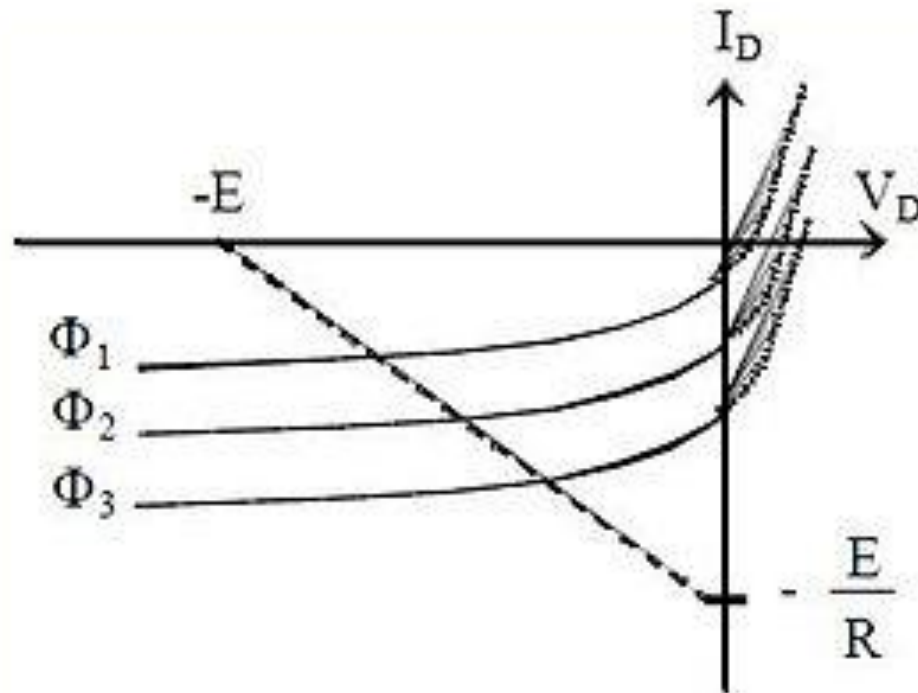
- ❑ **Avantages de l'utilisation d'un LASER comme un véhicule d'information: Gain et directivité**
- ❑ **Nouvelle application : LIFI et Trading haute fréquences**
- ❑ **Limitations du a la nécessité de presence de photorécepteurs**

Annexe1: Photodiode



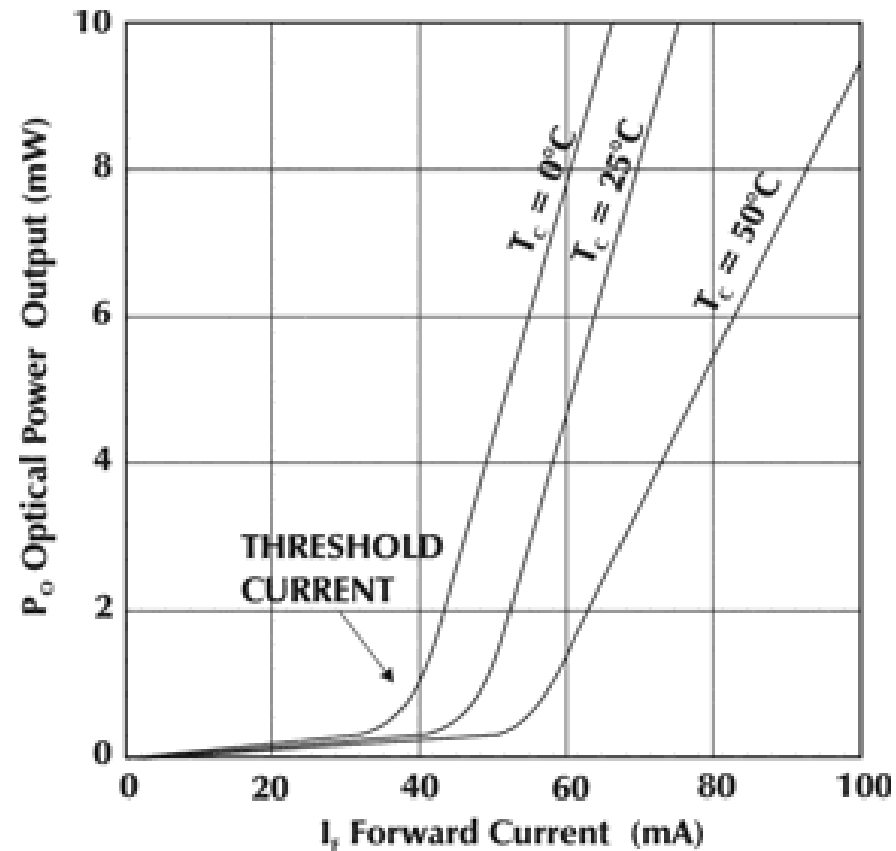
ANNEXE 2

Caractéristique de transfert



ANNEXE 3

Caractéristique de la diode laser photodiode





ANNEXE 4

**Le 1er code
python utilisés
est:**


```

1 # -*- coding: utf-8 -*-
2 """
3 Auteur : mehdi ben alaya
4
5
6
7 Ce programme commence par définir un paquet d'onde comme la somme de fonctions sinusoïdales.
8 Il permet de visualiser l'effet de différentes relations de dispersion sur la propagation de ces paquets d'onde.
9 Les différentes relations de dispersion utilisées sont :
10 - vide :  $k = \omega/c$ 
11 - milieu à coupure (plasma) :  $k = \sqrt{\omega^2 - \omega_c^2}/c$ 
12 - milieu transparent (indice optique) :  $k = (\omega/c) \cdot (1 + B \cdot \omega^2)$ 
13 - onde d'une particule non relativiste libre  $k = \sqrt{2 \cdot m \cdot \omega / \hbar}$ 
14
15 Le programme prend  $c = 1$ ,  $m = 1$ ,  $\hbar = 1$ . Les valeurs prises sont adimensionnées.
16
17 Le programme permet de tracer :
18 - le suivi d'un paquet d'onde pour observer son glissement de phase et son étalement
19 ATTENTION : si la simulation dure trop longtemps on verra les interférences entre le paquet d'onde suivi et les paquets voisins
20 - la propagation de plusieurs paquets d'onde dans le milieu dispersif
21
22
23 Les paramètres se changent à partir de la ligne 127.
24
25
26 """
27
28
29
30 import numpy
31 import math
32 from scipy.signal import get_window
33 import matplotlib.pyplot as plt
34 import matplotlib.animation as animation
35
36 class Onde:
37     def __init__(self, suivi=True, dispersion='vide', coupure=0.0, coeff=1.0):
38         """

```

```

: param suivi : suivi du paquet d'onde a la vitesse de groupe
: param dispersion : relation de dispersion
: param coupure : frequence de coupure pour la relation de dispersion 'coupure'
: param coeff : coefficient pour la relation de dispersion 'optique'
"""
self.freq = [1.0,3.0,5.0]
self.amp = [1.0,0.5,0.2]
self.phase = [0.0,0.0,0.0]
self.nf = len(self.freq)
if suivi:
    self.suivi = 1.0
else:
    self.suivi = 0.0
self.coupure = coupure
self.coeff = coeff
self.wc2 = (2*math.pi*coupure)**2
self.a = 1.0

if dispersion=='vide':
    self.k = self.k_vide
    self.vg = self.vg_vide

elif dispersion=='coupure':
    self.k = self.k_coupure
    self.vg = self.vg_coupure

elif dispersion=='optique':
    self.k = self.k_optique
    self.vg = self.vg_optique

elif dispersion=='debroglie':
    self.k = self.k_debroglie
    self.vg = self.vg_debroglie

k_vide(self,w):
return w
vg_vide(self,w):

```

```

def vg_vide(self,w):
    return 1.0

def k_coupure(self,w):
    return math.sqrt(w*w-self.wc2)

def vg_coupure(self,w):
    return math.sqrt(1-self.wc2/(w*w))

def k_optique(self,w):
    return w*(1.0+self.coeff*w*w)

def vg_optique(self,w):
    return 1.0/(1.0+3*self.coeff*w*w)

def k_debroglie(self,w):
    return math.sqrt(w)

def vg_debroglie(self,w):
    return 2.0*math.sqrt(w)

def paquet(self,f,P,window="hamming"):
    """
        Creation d'un paquet d'onde
        : param f : frequence centrale
        : param P : 2P+1 = nombre de frequences
        : param window : fenetrage
    """
    M = int(2*P+1)
    self.a = 2*math.pi*f/self.k(2*math.pi*f)
    self.freq = numpy.zeros(M)
    self.amp = numpy.zeros(M)
    self.phase = numpy.zeros(M)
    self.nf = M
    self.vgroupe = self.vg(2*math.pi*f)/self.a
    for n in range(M):
        self.freq[n] = f-P+n
        self.amp[n] = 1.0
        self.phase[n] = 0.0
    self.amp = get_window(window,M)

```



```

112 def echantillons(self,xmin,xmax,t,N):
113     """
114     calcul de N echantillons de l'onde reelle sur l'intervalle [xmin,xmax] a l'instant t
115     """
116     x = numpy.linspace(xmin,xmax,N)
117     y = numpy.zeros(x.size)
118     for i in range(self.nf):
119         w = 2*math.pi*self.freq[i]
120         k = self.k(w)*self.a
121         phi = k*x+(k*self.vgroupe*self.suivi-w)*t+self.phase[i]
122         y += self.amp[i]*numpy.cos(phi)
123     return (x,y)
124
125
126 def echantillons_proba(self,xmin,xmax,t,N):
127     """
128     calcul de N echantillons de la densite de probabilite sur l'intervalle [xmin,xmax] a l'instant t
129     """
130     x = numpy.linspace(xmin,xmax,N)
131     yr = numpy.zeros(x.size)
132     yi = numpy.zeros(x.size)
133     for i in range(self.nf):
134         w = 2*math.pi*self.freq[i]
135         k = self.k(w)*self.a
136         phi = k*x+(k*self.vgroupe*self.suivi-w)*t+self.phase[i]
137         yr += self.amp[i]*numpy.cos(phi)
138         yi += self.amp[i]*numpy.sin(phi)
139     return (x,numpy.absolute(yr+1j*yi))
140
141
142

```

```

143 #####
144 #####
145 ##### DEFINITION DES PARAMETRES #####
146 #####
147 #####
148
149 #choix de la frequence centrale du paquet d'onde

```

```

#choix de la relation de dispersion :
# 'vide' : propagation d'une onde électromagnétique dans le vide
# 'coupure' : relation de dispersion de type Klein-Gordon, choix de la fréquence de coupure avec le paramètre f_c
# 'optique' : relation de dispersion de Cauchy (dispersion dans un diélectrique, (par exemple le verre)), choix du coefficient B de  $n^2 = A + B/(\omega^2 - \omega_c^2)$ 
# 'debroglie' : relation de dispersion de l'équation de Schrödinger pour une propagation dans le vide (une vitesse de simulation de 0.4)

disp="coupure"
f_c=40
coefficient=0.1

#choix de la vitesse de simulation (par défaut 1.0)

vitesse = 1.0

#choix du tracé de :
# False => l'amplitude de la fonction d'onde/du champ électrique
# True => la densité de probabilité/de l'amplitude du vecteur de Poynting

intensite = False

#####
#####
##### Tracé des animations de dispersion #####
#####
#####

onde = Onde(suivi=True,dispersion=disp,coupure=f_c,coeff=coefficient*10**(-6))
onde2 = Onde(suivi=False,dispersion=disp,coupure=f_c,coeff=coefficient*10**(-6))

```

```

187 onde.paquet(frequence,20>window='hamming')
188 onde2.paquet(frequence,20>window='hamming')
189
190
191 temps = 0.0
192 dt = 0.005*vitesse
193 N = 5000
194 xmin = -0.35
195 xmax = 0.35
196
197 xmin2 = 0.
198 xmax2 = 4.0
199
200
201 if not intensite :
202
203     (x,y) = onde.echantillons(xmin,xmax,temps,N)
204     (x2,y2) = onde2.echantillons(xmin2,xmax2,temps,N)
205
206     fig, ((ax1, ax2)) = plt.subplots(2, 1)
207     line, = ax1.plot(x,y)
208     ax1.grid()
209     ax1.set_xlabel("Distance au centre du paquet d'onde (1)",fontsize=18)
210     ax1.set_ylabel("Amplitude (1)",fontsize=18)
211
212     line2, = ax2.plot(x2,y2)
213     ax2.grid()
214     ax2.set_xlabel("Distance dans le milieu dispersif (1)",fontsize=18)
215     ax2.set_ylabel("Amplitude (1)",fontsize=18)
216
217
218 def animate(i):
219     global temps,xmin,xmax,N
220     temps += dt
221     (x,y) = onde.echantillons(xmin,xmax,temps,N)
222     line.set_xdata(x)
223     line.set_ydata(y)
224     return line

```



```

global temps,xmin,xmax,N
temps += dt
(x2,y2) = onde2.echantillons(xmin2,xmax2,temps,N)
line2.set_xdata(x2)
line2.set_ydata(y2)
return line,line2

```

```

ani = animation.FuncAnimation(fig,animate,1000,interval=40),animation.FuncAnimation(fig,animate2,1000,interval=40)

```

e :

```

(x,y) = onde.echantillons_proba(xmin,xmax,temps,N)
(x2,y2) = onde2.echantillons_proba(xmin2,xmax2,temps,N)

fig, ((ax1, ax2)) = plt.subplots(2, 1)
line, = ax1.plot(x,y)
ax1.grid()
ax1.set_xlabel("Distance au centre du paquet d'onde (1)",fontsize=18)
ax1.set_ylabel("Amplitude (1)",fontsize=18)

line2, = ax2.plot(x2,y2)
ax2.grid()
ax2.set_xlabel("Distance dans le milieu dispersif (1)",fontsize=18)
ax2.set_ylabel("Amplitude (1)",fontsize=18)

```

```

def animate(i):
    global temps,xmin,xmax,N
    temps += dt
    (x,y) = onde.echantillons_proba(xmin,xmax,temps,N)
    line.set_xdata(x)
    line.set_ydata(y)
    return line,

```

260

261 `def animate2(i):`262 `global temps,xmin,xmax,N`263 `temps += dt`264 `(x2,y2) = onde2.echantillons_proba(xmin2,xmax2,temps,N)`265 `line2.set_xdata(x2)`266 `line2.set_ydata(y2)`267 `return line,line2`

268

269 `ani = animation.FuncAnimation(fig,animate,1000,interval=40),animation.FuncAnimation(fig,animate2,1000,interval=40)`

270

271 `plt.show()`

272



ANNEXE 5

**Le 2ème
code python
utilisés est:**


```

1#!/usr/bin/env python3
2# -*- coding: utf-8 -*-
3"""
4Mehdi Ben Alaya
5"""
6import numpy as np
7import matplotlib.pyplot as plt
8from sklearn.linear_model import LinearRegression
9
10X = [5,15,25,35,45]
11Y = [0.771,0.640,0.462,0.320,0.252]
12
13X = np.array(X)
14Y = np.array(Y)
15X = X.reshape(5,1)
16
17model = LinearRegression()
18model.fit(X,Y)
19
20pente = model.coef_
21ordonnee_origine = model.intercept_
22modele_theorique = [pente*x + ordonnee_origine for x in X]
23
24plt.plot(X,Y,"rp")
25plt.plot(X,modele_theorique,"k--")
26plt.xlabel("")
27plt.ylabel("")
28
29plt.title("")
30plt.grid()
31plt.show()
32
33print("La pente est ", pente)
34print("L'ordonnée à l'origine est", ordonnee_origine)
35
36

```