

Compte rendu : TP Réseau d'Accès Radio

Quentin Goulas, Lucas Hocquette

This report presents the scripts used and results obtained on the Lab of the Wireless Communications class, focusing on the notions of receivers. Aside from additional comments, all reported scripts are identical to the scripts provided in the Matlab Live Script document provided as support. All plots are direct outputs from each script.

Ce rapport présente les scripts utilisés et les résultats obtenus durant le Travail Pratique de Réseau d'Accès Radio. Mis à part des commentaires additionnels, tous les scripts présentés dans ce document sont identiques aux scripts fournis dans le Jupyter Notebook en guise de support. Tous les graphes sont des outputs directs des scripts fournis.

Librairies requises

```
import numpy as np
import matplotlib.pyplot as plt
```

1 Capacité d'un système CDMA

1 On définit la fonction `sample_users` qui prend en entrée un nombre d'utilisateurs K et un rayon R , et retourne les positions de K utilisateurs uniformément répartis dans un cercle de rayon R .

```
def sample_users(K,R):
    v = np.random.uniform(low=0,high=R**2,size=K)
    theta = np.random.uniform(low=0,high=2*np.pi,size=K)
    r = np.sqrt(v)
    x,y = r*np.cos(theta), r*np.sin(theta)
    return x,y
```

2

```
r = 1
W = 3.84*10**6
theta = 0.4
sigma2 = 10**(-104/10)*1e-3
P = 10**(-40/10)*1e-3

def measure_achievement_ratio(K,R,gamma,n_avg=1):
    x,y = sample_users(K,r)
    d = np.sqrt(x**2+y**2)
    L = -128.1 - 37.6*np.log10(d)
    l = 10**(L/10)
    history = np.zeros(n_avg)
    for i in range(n_avg):
        h = np.random.exponential(0.5,K)
```

```
16     g = l*h
17     p = P/K
18     SINR = W/R*p*g/(theta*(K-1)*p*g+sigma2)
19     history[i] = np.mean(SINR>=gamma)
20
21     return history
22
23 print(f'The percentage of users for which the decoding condition is satisfied is : {
24       measure_achievement_ratio(20,32*1e3,10**(7/10))[0]*100}%')
```

Avec la configuration donnée, 100% des utilisateurs satisfont la condition.

3 et 4

```
2 achievement_ratio = measure_achievement_ratio(20,32*1e3,10**(7/10),100)
3 print(f'delta = {np.mean(achievement_ratio)*100}%')
```

On obtient des résultats de 98.4%.

5

```
7 K_values = np.array(range(1,100))
8 deltas = np.zeros(len(K_values))
9
10 for i,K in enumerate(K_values) :
11     ach_rat = measure_achievement_ratio(K,32*1e3,10**(7/10),100)
12     deltas[i] = np.mean(ach_rat)
13
14 print(deltas)
15 print(f'The maximum number of users on the network is {K_values[np.sum(deltas>=0.9)]} ')
16
17 plt.plot(K_values,deltas)
18 plt.show()
```

Le nombre maximal d'utilisateurs obtenu est 44.

2 Contrôle de puissance uplink d'un système CDMA : capacité et solutions itératives

3 Comparaison entre systèmes CDMA et TDMA