


git clone https://gitlab.osupytheas.fr/quentin.c/atelier_radar.git

Préambule

Ce TP est construit à partir des supports de cours de S.J. Orfanidis, "Electromagnetic Waves and Antennas" à la Rutgers University (2016) à consulter gratuitement sur <https://www.ece.rutgers.edu/~orfanidi/ewa/>

 python™ Les scripts vous sont proposés en langage Python.

Mais ne vous inquiétez pas, pas besoin d'installation. Tout est écrit dans un Jupyter Notebook.

Si vous n'avez pas d'interpréteur python, ou Jupiter Notebook, je vous invite à ouvrir depuis votre navigateur: <https://jupyter.org/try-jupyter>

Bon TP !

Diagramme de rayonnement et directivité d'un réseau d'antennes

Quelques notions théoriques

Le diagramme de rayonnement

Le diagramme représente les variations de la puissance rayonnée par l'antenne dans les différentes directions de l'espace. Il varie entre 0 et 1 selon la direction. Celui-ci peut se représenter sous différentes formes de représentations dans l'espace (cartésien, polaire, ou sphérique). En coordonnées sphériques, les variables spatiales utilisées sont :

- r la distance au point de mesure
- ϕ est l'angle dans le plan horizontal (azimuth),
- θ l'angle dans le plan vertical (élévation).

```
In [1]: #

1) Faites varier dans le code suivant la distance  $d$  séparant deux éléments, et le facteur de phase  $w$  entre la deuxième et la première antenne. Tester les valeurs particulières  $d = \lambda, \lambda/2, \lambda/4$ , et des excitations d'antennes d'intensité égale mais avec différents déphasages.

Dans la configuration "broadside" le faisceau est dirigé perpendiculairement au réseau linéaire d'antennes, soit à  $90^\circ$  ou selon l'axe des ordonnées. Dans la configuration "endfire", le faisceau est dirigé dans le même axe que le réseau d'antennes.

2) Pour un réseau d'émission à deux antennes, quelle est la configuration qui permet d'illuminer au mieux  $180^\circ$  de l'espace ?

3) Comment orientera-t-on le réseau d'émission par rapport au réseau de réception (cas d'une émission/réception simultanée) ?

```
In [6]: # lam : longueur de l'onde monochromatique
d : espacement entre chaque element d'antenne
w : facteur de phase
la première antenne sert de référence son poids est à 1
le facteur de phase est donnée en nombre complexe
pour une excitation de même amplitude sur les 2 antennes
on prend delta pour le déphasage et w=
1 correspond à un retard nul, soit delta=0
1j pour une quadrature de phase positif, soit delta=pi/2
-1 pour une opposition de phase, soit delta=pi
-1j pour une quadrature de phase negatif, soit delta=-pi/2

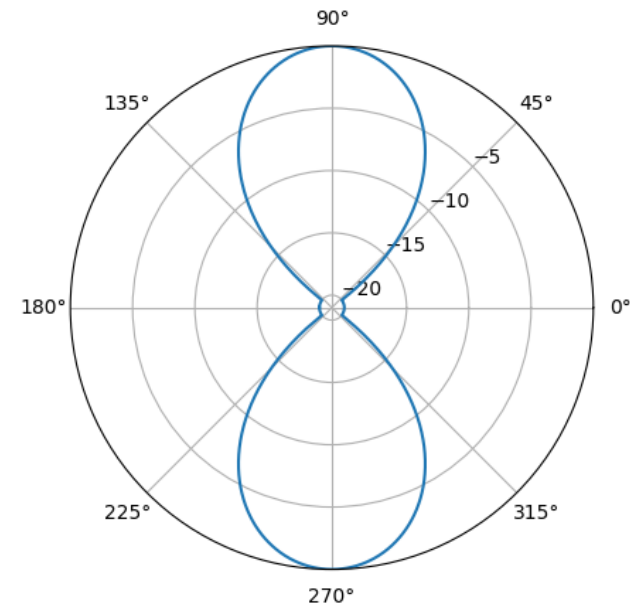
lam : la longueur de l'onde monochromatique du champ
lam= 1

'''faites varier d pour obtenir différents diagramme'''
d: l'espace entre les antennes,
proportionnel à 'lam' la longueur d'onde
d = lam/2

''' idem pour a1, a2, et delta'''
2 antennes avec la meme excitation (amplitude et phase)
a1=1
a2=1
delta=0.*np.pi
w = [a1,a2]*np.array([1, np.cos(delta)+1j*np.sin(delta)])

Calculate gain and directive gain;
phi, AFN, gN = directivity(d, w)
DNdBi = get_directive_gain(gN)

plot on a polar chart.
plt.polar(phi, DNdBi)
ax = plt.gca()
ax.set_rticks([-20, -15, -10, -5])
ax.set_rlabel_position(45)
plt.show()
```



## 2 ) Ouverture d'un réseau linéaire à N antennes

Etudions l'arrivée d'une onde électromagnétique sur un réseau de  $N$  antennes alignées le long d'une ligne droite et espacées d'une distance constante  $d$ . L'angle entre la perpendiculaire au réseau et l'onde électromagnétique est  $\theta$ .

1. Faites varier le nombre d'antennes  $N$  pour une longueur de réseau donnée  $L$ .  
Que remarquez-vous ?
2. Attention, à l'espace minimum entre deux antennes. Quel est le nombre d'antennes maximum pour éviter les interactions entre antennes ?
3. Faites varier la longueur  $L$  du réseau. Que remarquez-vous ?

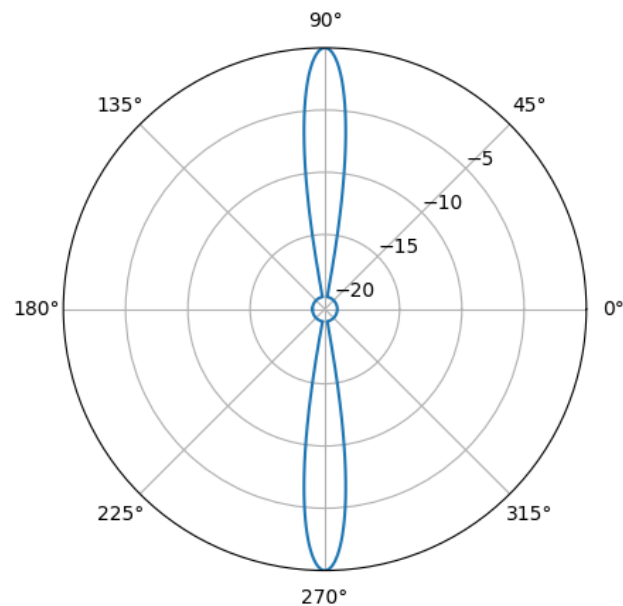
```
In [7]: lam = 1/2
L = 4.*lam

'''N : nombre d'éléments'''
N = 8
'''N : nombre d'éléments'''
d = L/N

amplitude = np.full(N,1)
phase = np.full(N,0.*np.pi)
w = np.multiply(amplitude,np.exp(1j*phase))

Calculate gain and directive gain;
phi, AFN, gN = directivity(d, w)
DNdBi = get_directive_gain(gN)

plot on a polar chart.
plt.polar(phi, DNdBi)
ax = plt.gca()
ax.set_rticks([-20, -15, -10, -5])
ax.set_rlabel_position(45)
pour passer en coordonnées géographiques
'''
ax.set_theta_zero_location('N') # make 0 degrees point up
ax.set_theta_direction(-1) # increase clockwise
'''
plt.show()
```



### 3) Direction d'arrivée d'une source (ou plusieurs sources)

La formation de voies (ou goniométrie, en anglais "beamforming") est une technique applicable à un groupement d'antennes dans le but de localiser la provenance de signaux incidents ou angles d'arrivées ("Direction of Arrivals" – DOA). Le réseau d'antenne opère donc en mode réception. Elle est équivalente à un balayage électronique qui regarde dans la zone limite qui constitue la fenêtre d'observation. Le résultat de la formation de voies est un pseudo-spectre contenant des pics identifiant les angles d'arrivées.

Le signal reçu par la k-ième antenne se met sous la forme:

$$y(\mathbf{k}) = \mathbf{a}\phi s(\mathbf{k}) + n(k)$$

avec  $\mathbf{a}\phi$  le vecteur phaseur (steering vector) du réseau.

Le pseudo-spectre dont les pics maximum correspondent aux directions des sources s'écrit sous la forme:

$$BF = \mathbf{a}^H R \mathbf{a}$$

avec  $\cdot^H$  l'opérateur hamiltonien et  $R$  la matrice de covariance.

Le code utilisé est celui issu du "PySDR: A Guide to SDR and DSP using Python" par Dr. Marc Lichtman <https://pysdr.org/content/doa.html>

- 1) Faites varier le nombre d'antennes N, la direction d'arrivée, ou le niveau de bruit
- 2) Remarquez vous des angles limites pour la direction d'arrivée ?
- 3) Ajoutez une deuxième source avec une autre direction d'arrivée.

```

In [10]: # on defini le réseau d'antennes ULA
N=8
lam=1
d=lam/2
amplitude = np.full(N,1)
phase = np.full(N,0.*np.pi)
w = np.multiply(amplitude,np.exp(1j*phase))

le signal émis depuis une source lointaine est de la forme
''' MHertz '''
sample_rate = 1e9
''' number of time elements '''
Nsamples = 60*60
t = np.arange(Nsamples)/sample_rate # time vector
''' light speed'''
c = 3. * 10**8
''' frequency '''
f_tone = 0.05* c/lam

s=100*np.exp(2j * np.pi * f_tone * t)

la direction d'arrivée du signal
theta0_deg=30
theta0=np.deg2rad(theta0_deg)

on applique le retard à chaque antenne
a = np.exp(1j * 2*np.pi * d * np.arange(N) * np.cos(theta0))

chaque antenne recoit un signal avec un retard
donné pour le facteur réseau
A = np.asmatrix(a)
S = np.asmatrix(s)

la matrice des signaux recus (par antennes, par pas de temps)
.T matrix transpose
@ matrix multiplication
R = A.T @ S

bruit - signal aleatoire de petite amplitude (noise)
n = 5 * np.random.randn(N, Nsamples) + 1j*np.random.randn(N, Nsamples)
R = R + n

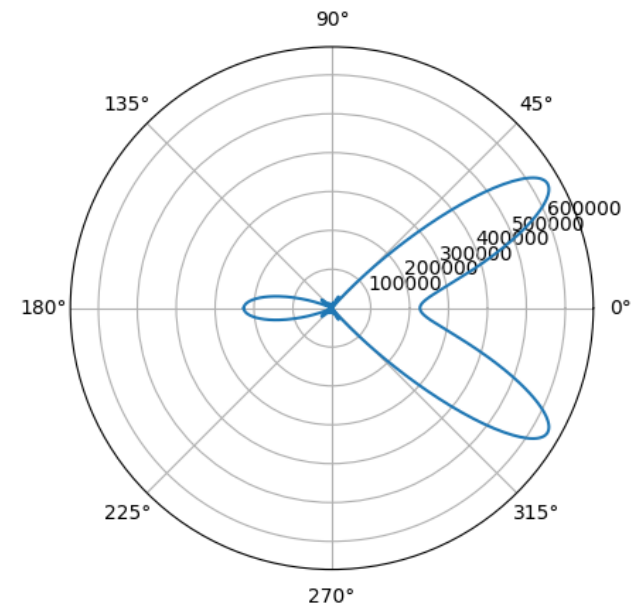
on utilise toutes les directions d'arrivées possibles
theta_scan = np.linspace(-1*np.pi, np.pi, 1000) # 1000 different thetas b
results = []
for theta_i in theta_scan:
 #print(theta_i)
 w = np.asmatrix(np.exp(2j * np.pi * d * np.arange(N) * np.cos(theta_i)
 r_weighted = np.conj(w) @ R # apply our weights corresponding to the
 r_weighted = np.asarray(r_weighted).squeeze() # get it back to a norm
 results.append(np.mean(np.abs(r_weighted)**2)) # energy detector

print angle that gave us the max value
print("theta pour max(DOA) :",abs(theta_scan[np.argmax(results)]) * 180 /

fig, ax = plt.subplots(subplot_kw={'projection': 'polar'})
ax.plot(theta_scan, results)
plt.show()

```

theta pour max(DOA) : 30.09009009009011



Pour compléter cette étude, voici un lien vers l'étude des différentes méthodes d'implémentation des estimateurs de Direction Of Arrivals : [https://github.com/petotamas/pyArgus/blob/master/docs/nb\\_direction\\_of\\_arrival\\_estimator.ipynb](https://github.com/petotamas/pyArgus/blob/master/docs/nb_direction_of_arrival_estimator.ipynb)

In [ ]: