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/



puthon™ 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

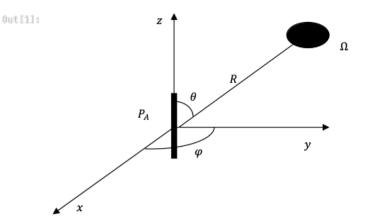
Le diagramme de rayonnement

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 shérique). En coordonnées sphériques, les variables spatiales utilisées sont :

- \$r\$ la distance au point de mesure
- \$\phi\$ est l'angle dans le plan horizontal (azimuth),
- \$\theta\$ l'angle dans le plan vertical (élévation).

In [1] #<img src="" alt="géométrie du diagramme de rayonnement" style="width:350 # ou ![géométrie du diagramme de rayonnement](figl_coordonnees.png) from IPython.display import Image Image(filename='fig1_coordonnees.png')

16/10/2023, 23:352 sur 14 16/10/2023, 23:35 1 sur 14



1. Puissance rayonnée par une antenne dans une direction de l'espace

La directivité d'une antenne est définie comme le rapport de l'intensité de rayonnement dans une direction donnée (θ, ϕ) par rapport à l'intensité (puissance) de rayonnement moyennée sur toutes les directions. L'intensité de rayonnement moyenne est égale à la puissance totale rayonnée par l'antenne divisée par 4π .

L'expression générale pour la directivité s'écrit en fonction de l'intensitté de rayonnement $F(\theta, \phi)$ comme:

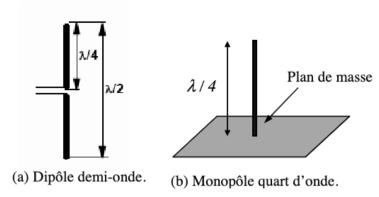
$$D(heta,\phi) = 4\pi * rac{F(heta,\phi)}{\int_0^{2\pi} \int_0^{\pi} F(heta,\phi) \sin heta d heta d\phi}$$

L'antenne élémentaire

On travaillera tout d'abord avec des éléments d'antennes élémentaires. On peut prendre par exemple des antennes filaires simples comme le dipole $\lambda/2$ ou le monopole $\lambda/4$

In [2]: #<img src="fig2 antennesElementaires.png" alt="antennes élémentaires" sty Image(filename='fig2 antennesElementaires.png')

Out[2]:



a) Dipôle demi-onde

C'est l'antenne filaire rectiligne historiquement la plus ancienne mais toujours très utilisée. Elle est constituée de deux brins métalliques alignées sur le même axe et alimentée en son centre (fig. 2a). Les deux autres extrémités libres des deux brins sont laissées ouvertes, ce qui permet d'imposer une valeur nulle du courant en ces points. On aboutit ainsi à une répartition de type onde stationnaire avec un maximum de courant à l'entrée du dipôle. Encore appelée « antenne fouet », est composée dans sa fonction réceptrice d'un conducteur cylindrique perpendiculaire à un plan métallique prenant des dimensions proches du mètre. Le monopole ainsi constitué permet la mesure de champs électriques polarisés parallèlement à l'axe du conducteur. Un adaptateur d'impédance branché à la base du conducteur s'avère généralement indispensable dès que la fréquence des signaux procure des longueurs d'onde bien plus grandes que la dimension de l'antenne.

En champ lointain, le rayonnement autour de l'axe du conducteur est identique dans toutes les directions ϕ et ne dépend que de l'angle d'élévation θ . L'antenne dipôle est omnidirectielle dans le plan horizontal $\theta=90^\circ$.

Le diagramme de rayonnement du dipole de longueur $L=\lambda/2$ est :

$$D(heta) = rac{cos(rac{\pi}{2}cos(heta))}{sin(heta)}$$

b) Antenne monopole

Encore appelée « antenne fouet », est composée d'un fil (ou fouet) conducteur d'une longueur égale au quart de l'onde monchromatique étudiée et placé perpendiculaire à un plan de masse (si possible métalique). Le monopole ainsi constitué permet la mesure de champs électriques polarisés parallèlement à l'axe du conducteur.

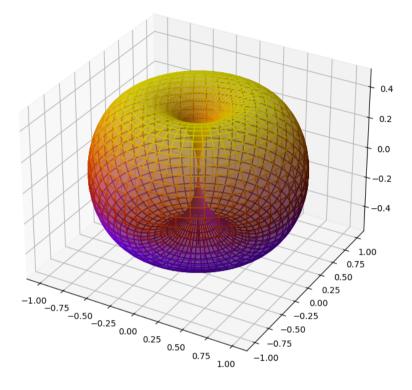
Pour le monopole, le diagramme de rayonnement est identique à celui du dipole au dessus du plan de masse, et nul dans la partie inférieure.

3 sur 14 16/10/2023, 23:354 sur 14 16/10/2023, 23:35

```
In [3] # on importe des bibliothèques de programmes python
        import numpy as np
        import matplotlib.pyplot as plt
        import mpl_toolkits.mplot3d.axes3d as axes3d
        import matplotlib.colors as mcolors
         DEFINITION DE LA FONCTION sph2cart
         de passage de coordonnnees spheriques a cartesiennes
        def sph2cart(azimuth,elevation,r):
            x = r * np.sin(elevation) * np.cos(azimuth)
            y = r * np.sin(elevation) * np.sin(azimuth)
            z = r * np.cos(elevation)
            return x, y, z
        111
         PROGRAMME
        # initialisation des vecteurs de coordonnees spheriques
        # discretisation de l'angle d'elevation de 0 à pi
        # et de l'azimuth de 0 à 2*pi
        theta = np.linspace(0.001, np.pi, 400)
        phi = np.linspace(0.001,2*np.pi,400)
        # mises sous forme matricielle
        THETA, PHI = np.meshgrid(theta,phi)
        # calcul du diagramme de rayonnement
        D = np.cos(np.pi/2*np.cos(THETA))/np.sin(THETA)
        # calcul des coordonnnees cartesiennes
        X, Y, Z = sph2cart(PHI,THETA,D)
        # mise en forme de la figure
        fig = plt.figure(figsize=(7,7))
        cmap = plt.get cmap('qnuplot')
        ax = plt.subplot(111, projection='3d')
        norm = mcolors.Normalize(vmin=Z.min(), vmax=Z.max())
        ''' le diagramme de rayonnement'''
        ax.plot_surface(X, Y, Z, rstride=8, cstride=8,
                         facecolors=cmap(norm(Z)), antialiased=True, alpha=0.5)
        '''le dipole'''
        ax.scatter3D(0.0.0)
        fig.suptitle("Diagramme de rayonnement d'un dipole (en 3D)")
        plt.tight_layout()
        plt.show()
```

16/10/2023, 23:35

Diagramme de rayonnement d'un dipole (en 3D)



Le diagramme d'un réseau

Un réseau d'antennes peut être utilisé pour diriger les ondes radio dans une direction particulière en ajustant leur nombre, leur disposition géométrique ainsi que leurs amplitudes et phases relatives. Nous allons mettre en pratique les différentes équations d'électromagnétisme pour différentes configurations, et simuler le comportement directif des réseaux ainsi créés.

Pour déterminer les caractéristiques de rayonnement en champ lointain du réseau $F_N(\theta,\phi)$, il est nécessaire de calculer le facteur de réseau de l'antenne réseau noté AF et de multiplier la fonction caractéristique de rayonnement d'un élément rayonnant $f(\theta,\phi)$ par le facteur de réseau $AF(\theta,\phi)$.

Ce calcul suppose que les interactions entre les antennes soient négligeables, c'est à dire que le rayonnement de chaque antenne n'est pas perturbé par la présence des autres antennes. La zone du champ lointain pour une antenne 1/4 d'onde est atteinte pour une distance $d>>\frac{2L^2}{\lambda}$, soit $d>>\frac{\lambda}{8}$ (voir C. A. Ballanis, Antenna Theory : Analysis and Design, Fourth Edition, 2016 John Wiley & sons").

Si une seule antenne produit un vecteur de rayonnement $f(\theta,\phi)$, alors le vecteur de rayonnement total F_N dû à l'ensemble des N antennes est :

$$F_N(\theta,\phi) = AF(\theta,\phi)f(\theta,\phi)$$

Dans le cas d'un réseau de N antennes omnidirectionnelles placées arbitrairement dans le plan (xy), on utilisera les coordonnées (x_j,y_j) des antennes pour calculer le champ rayonné.

$$F_N(heta,\phi) = f(heta,\phi) \sum_{j=0}^{N-1} I_j e^{i.lpha_j} e^{i(k.x_j sin(heta)cos(\phi) + k.yy_j sin(heta)sin(\phi))}$$

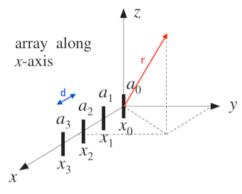
Le réseau linéaire équidistant :

Considérons un ensemble de N antennes omnidirectionnelles alignées le long de l'axe x et espacées régulièrement de d. On placera le centre du repère de calcul sur la première antenne, qui sera donc située à $x_0=0$, et la dernière antenne à $x_{N-1}=(N-1)*d$. u, avec u le vecteur unitaire sur l'axe des abscisses (x).

In [4]: #<img src="uniformLinearAntennaArray.png" alt="linear array" class="bg-p
Image(filename='uniformLinearAntennaArray350px.png')</pre>

5 sur 14 16/10/2023, 23:356 sur 14

Out[4]:



Si une seule antenne produit un vecteur de rayonnement f(k), où $k=rac{2\pi}{\lambda}r$, le vecteur de rayonnement total dû à l'ensemble de N antennes est

$$F_N((k))=\sum_{i=0}^{N-1}w_je^{ik\cdot d_j}f(k)=AF(k)F(k)$$

où w_j est le coefficient d'alimentation de la $j^{\grave{e}me}$ antenne, représentant son amplitude et sa phase.

Nous pouvons choisir $w_0=1$ pour spécifier les coefficients d'alimentation par rapport à l'antenne à l'origine.

Si nous choisissons en outre de ne considérer que la contribution azimutale (ϕ) au rayonnement dans le plan xy, en fixant l'angle d'élévation θ à 90° ou $\pi/2$, nous avons :

$$AF(\phi) = \sum_{i=0}^{N-1} w_j e^{i.j.k\cdot d.cos(\phi)}$$

soit en utilisant la relation entre le nombre d'onde k et la longueur d'onde λ :

$$AF(\phi) = \sum_{i=0}^{N-1} w_j e^{i \cdot rac{2.\pi}{\lambda} \cdot j \cdot d.cos(\phi)}$$

La directivité (relatif à la puissance de rayonnement) est le carré de cette quantité.

7 sur 14 16/10/2023. 23:358 sur 14

```
In [5] # fonction de gain en puissance
        # depend de "d" l'espace entre les antennes
        # et "w" poids (amplitude, phase) de chaque antenne
        def directivity(d, w):
            phi = np.linspace(-1*np.pi, np.pi, 1000)
            '''Return the power as a function of azimuthal angle, phi'''
            psi = 2*np.pi * d / lam * np.cos(phi)
            ''' facteur de reseau pour N antennes equidistantes'''
            for i in range(len(w)-1):
                AF = AF + w[i+1]*np.exp(1i*(i+1)*psi)
            qD = np.abs(AF)**2
            return phi, AF, gD
        # fonction de gain directif
        def get directive gain(gD, minDdBi=-20):
            """Return the "directive gain" of the antenna array producing gain g.
            DdBi = 20 * np.log10(gD / np.max(gD))
            return np.clip(DdBi, minDdBi, None)
```

Cas d'étude

1) réseau de 2 antennes à l'émission

Pour deux antennes identiques, on a :

$$D(\phi)=\left|AF(\phi)
ight|^2=\left|w_0+w_1.\,e^{i.k.d.cos(\phi)}
ight|^2$$

Le code suivant modélise le diagramme de rayonnement pour deux dipoles et trace en decibel (dB), $20log10(D/D_{max})$, en fonction de ϕ pour le cas à deux antennes sur un graphique polaire. Il est issu des examples de programmation en python scientifique (scipy): https://scipython.com/book2/chapter-7-matplotlib/examples /modelling-an-antenna-array/

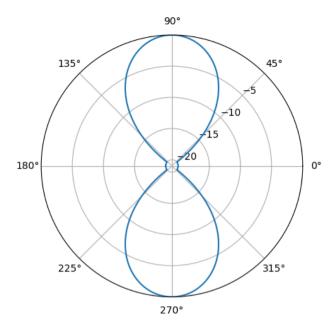
1) Faites varier dans le code suivant la distance d séparant deux éléments, et le facteur de phase w entre la deuxième et la première antenne. Tester les valeurs particulières $d=\lambda,\lambda/2,\lambda/4$, et des excitations d'antennes d'intensité égale mais avec différents déphasages.

Dans la configuration "broadside" le faisceau est dirigé perpendiculairement au réseau linéaire d'antennes, soit à 90° 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° de l'espace ?
- 3) Comment orientera-t-on le réseau d'émission par rapport au réseau de réception (cas d'une émission/réception simultanée) ?

16/10/2023, 23:35

```
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 donnnée en nombre complexe
        # pour une excitationn de même amplitude sur les 2 antennes
        # on prend delta pour le déphasage et w=
                1 correspond à un retard nul, soit delta=0
                li 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.qca()
        ax.set_rticks([-20, -15, -10, -5])
        ax.set_rlabel_position(45)
        plt.show()
```



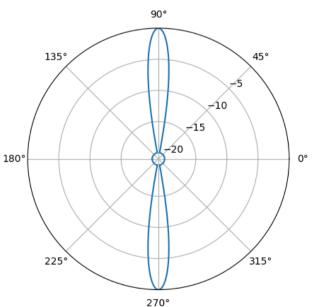
2) Ouverture d'un réseau linéaire à N antennes

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 perpendicutlaire au réseau et l'onde électromagnetique est θ .

- 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 remarguez-vous ?

16/10/2023, 23:35

```
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, qN = 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()
```



11 sur 14 16/10/2023. 23:3512 sur 14

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 $.^H$ 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

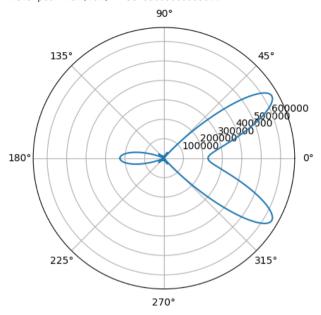
- 1) Faites varier le nombre d'antennees N, la direction d'arrivée, ou le niveau de bruit
- 2) Remarquez vous des angles limites pour la direction d'arrivée ? 3) Ajoutez une deuxième source avec une autre direction d'arrivée.

16/10/2023, 23:35

```
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
         ''' frequence '''
         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()
```

13 sur 14 16/10/2023. 23:3514 sur 14

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_estimaton.ipynb

]: