

Formation P.I.S, GE3, option Energies Renouvelables.

Cas d'étude éolienne

Florent Ravelet ^{*1} and Martin Bourhis¹

¹LIFSE - Arts et Métiers Sciences et Technologie, 151 boulevard de l'Hôpital, 75013 Paris, France

5 octobre 2021

1 Organisation

Ce projet se déroulera en quatre séances en groupe d'ED (er1/er2 d'une part, er3/er4 d'autre part) :

- Le 6 octobre 2021 en salle info ;
- Le 11 octobre 2021 en salle info ;
- Le 18 octobre 2021 en soufflerie pour les er1/er2 en salle info pour les er3/er4 ;
- Le 25 octobre 2021 en soufflerie pour les er3/er4 en salle info pour les er1/er2.

L'objectif est de :

1. concevoir via python une éolienne répondant à un cahier des charges donné (dimensions, vitesse du vent et rapidité spécifique au point de fonctionnement nominal), en utilisant différentes méthodes de dessin.
2. d'en faire la CAO sous freeCAD (nous l'imprimerons en impression 3D par dépôt de fil au labo)
3. de simuler numériquement l'écoulement autour de l'éolienne sous StarCCM+, pour plusieurs points de fonctionnement autour du nominal afin d'en esquisser la caractéristique
4. de réaliser des mesures de la courbe caractéristique (Coefficient de puissance fonction de la rapidité spécifique) en soufflerie, sur votre prototype

Vous formerez 4 groupes au sein de chaque ED (donc 8 groupes en tout, soit des groupes de 2 à 4 personnes). Chaque groupe se verra confier une méthode de dessin et une vitesse spécifique particulière.

2 Cahier des charges

Une éolienne de diamètre 300 mm a été conçue afin de récupérer un maximum de puissance sous un vent de 10 m.s^{-1} , à une vitesse spécifique ("tip-speed ratio") $\Lambda = 1$. La méthode de conception et la géométrie obtenue sont décrites dans l'article de Bourhis et al. [2022], en § 2, disponible sur SAVOIR.

Il s'agit pour vous de concevoir des éoliennes avec une méthode de conception identique, en variant quelques paramètres, et en respectant les contraintes suivantes :

- Diamètre extérieur de 100 mm (rayon en extrémité de pale $R_e = 50 \text{ mm}$).
- Rapport de moyeu de 0.3 (donc le rayon en pied de pale vaut $R_i = 15 \text{ mm}$).
- Vitesse du vent pour les tests 15 m.s^{-1} , avec une masse volumique $\rho = 1.2 \text{ kg.m}^{-3}$.
- Méthode de calcul des angles β_1 et β_2 identique à celle de Bourhis et al. [2022] (§ 2.2, équations 15-17).
- Une valeur de vitesse spécifique au point nominal de $\Lambda = 1$ pour les er1/er2 (comme dans l'article) et de $\Lambda = 0.75$ pour les er3/er4 (un peu plus basse).

*florent.ravelet@ensam.eu

Paramètres libres :

- Le nombre de pales.
- Les valeurs du serrage le long des pales.
- L'épaisseur relative des profils.

3 Travail préparatoire

- Lire l'article, au moins les § 1 et § 2.
- Implémenter la méthode (en python) pour aboutir aux lois $\beta_1(r)$ et $\beta_2(r)$ adaptées à votre cas.
- Calculer également la loi de corde des profils selon vos paramètres.

4 Outils

Pour le dimensionnement, vous avez sur SAVOIR un script python :

« `Python_FaitAubage_Grille.py` ».

Vous devez rentrer dans ce script aux lignes 53 à 58 les tableaux de rayons, angles d'entrée et de sortie, cordes et épaisseurs relatives. Ce script « fabrique » un fichier `csv` qu'on ouvrira dans un second temps avec un outil de CAO.

Références

M. Bourhis, M. Pereira, F. Ravelet, and I. Dobrev. Innovative design method and experimental investigation of a small-scale and very low tip-speed ratio wind turbine. *Experimental Thermal and Fluid Science*, 130 :110504, 2022.

Script

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 19 16:52:33 2018

@author: florent
"""

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import csv

# Fonction trace une ligne aux abscisses xx(0->1)
# de tangente en entree beta1 et en sortie beta2
# renvoie yy et l'angle courant de la tangente a la courbe

def lignemoy(xx,beta1,beta2):
    dyy=(np.tan(np.deg2rad(beta2))-np.tan(np.deg2rad(beta1)))*xx +
        np.tan(np.deg2rad(beta1))
    yy=(np.tan(np.deg2rad(beta2))-np.tan(np.deg2rad(beta1)))*np.power(xx,2)/2 +
        np.tan(np.deg2rad(beta1))*xx
    betai=np.rad2deg(np.arctan(dyy))
    return yy,betai

# Fonction d'habillage par un NACA00tt
# prend xx(0->1), yy(xx) l'equation de la ligne moyenne
# betai l'angle de la tangente a la courbe
# tt l'epaisseur relative
# l la corde

def epaisseur(xx,yy,betai,tt,l):
    naca=np.array([0.2980 , -0.1273 , -0.3506 , 0.2934 , -0.1005])
    yt=tt/0.2*(naca[0]*np.power(xx,0.5)+naca[1]*xx+naca[2]*np.power(xx,2)+
    naca[3]*np.power(xx,3)+naca[4]*np.power(xx,4))
    cc=np.power(np.power(xx[-1],2)+np.power(yy[-1],2),0.5)
    xp1=xx-cc*yt*np.sin(np.deg2rad(betai))
    yp1=yy+cc*yt*np.cos(np.deg2rad(betai))
    xp2=xx+cc*yt*np.sin(np.deg2rad(betai))
    yp2=yy-cc*yt*np.cos(np.deg2rad(betai))
    xp1=xp1/cc*l
    yp1=yp1/cc*l
    xp2=xp2/cc*l
    yp2=yp2/cc*l
    x=np.concatenate((xp1[::-1],xp2[1::]))
    y=np.concatenate((yp1[::-1],yp2[1::]))
    return x,y

#####
# MAIN
# rayons, angles beta1 et beta2, cordes, epaisseurs
# a remplacer si besoin par une lecture de fichier
# facteur 1000 pour la CAO mm
```

```

outfile='GSD_MultipaleElevesB.csv'
nr=11
rayons=np.linspace(50,150,nr) # en mm
beta1=np.array([65., 62., 59., 56., 53., 50., 47., 44., 41., 38., 35.]) # en deg
beta2=np.array([30. , 27.5, 25. , 22.5, 20. , 17.5, 15. , 12.5, 10. , 7.5,
5. ])# en deg
cordes=np.linspace(60,82.5,11)# en mm

epaisseurs=np.ones(nr)*0.1 # relatives (0.1 = 10%)

# vecteur ligne moyenne
xba=np.linspace(0.,0.03,num = 15)
xmc=np.linspace(0.031,0.2,num = 31)
xbf=np.linspace(0.21,1,num = 31)
xx=np.concatenate((xba, xmc, xbf))
npts=xx.size

# on ouvre un fichier en ecriture
with open(outfile, 'w') as csvfile:
    spamwriter = csv.writer(csvfile, delimiter='\t', lineterminator='\r')
    spamwriter.writerow(['StartLoft'])

# boucle sur les rayons
for i in range(nr):
    spamwriter.writerow(['StartCurve'])
    # ligne moyenne de x=0 a x=1
    # inclinee de beta1 a beta2
    lili=lignemoy(xx,beta1[i],beta2[i])
    yy=lili[0]
    betai=lili[1]
    # habillage avec un naca 00xx
    # epaisseur xx % "e"
    # mise a echelle avec corde "l"
    profil=epaisseur(xx,yy,betai,epaisseurs[i],cordes[i])
    # vue 2D
    plt.figure(1)
    plt.plot(profil[0],profil[1])

    # Passage en cylindrique
    # projection du profil sur le moyeu
    # angles de projection
    theta=profil[0]/rayons[i]
    ZZ=profil[1] # direction axiale
    YY=rayons[i]*np.sin(theta) # direction tangentielle
    XX=rayons[i]*np.cos(theta) # direction radiale

# vue a 3D
fig = plt.figure(2)
ax = fig.gca(projection='3d')
ax.plot(XX,YY,ZZ)
ax.set_xlabel('X -> r')
ax.set_ylabel('Y -> theta')

```

```
ax.set_zlabel('z -> z')
#plt.axis('equal')
# export du profil courant
for j in range(XX.size):
    spamwriter.writerow([XX[j] , YY[j] , ZZ[j]])
    spamwriter.writerow(['EndCurve'])
spamwriter.writerow(['EndLoft'])
spamwriter.writerow(['End'])
```