

VSP : Propulseur Voith-Schneider



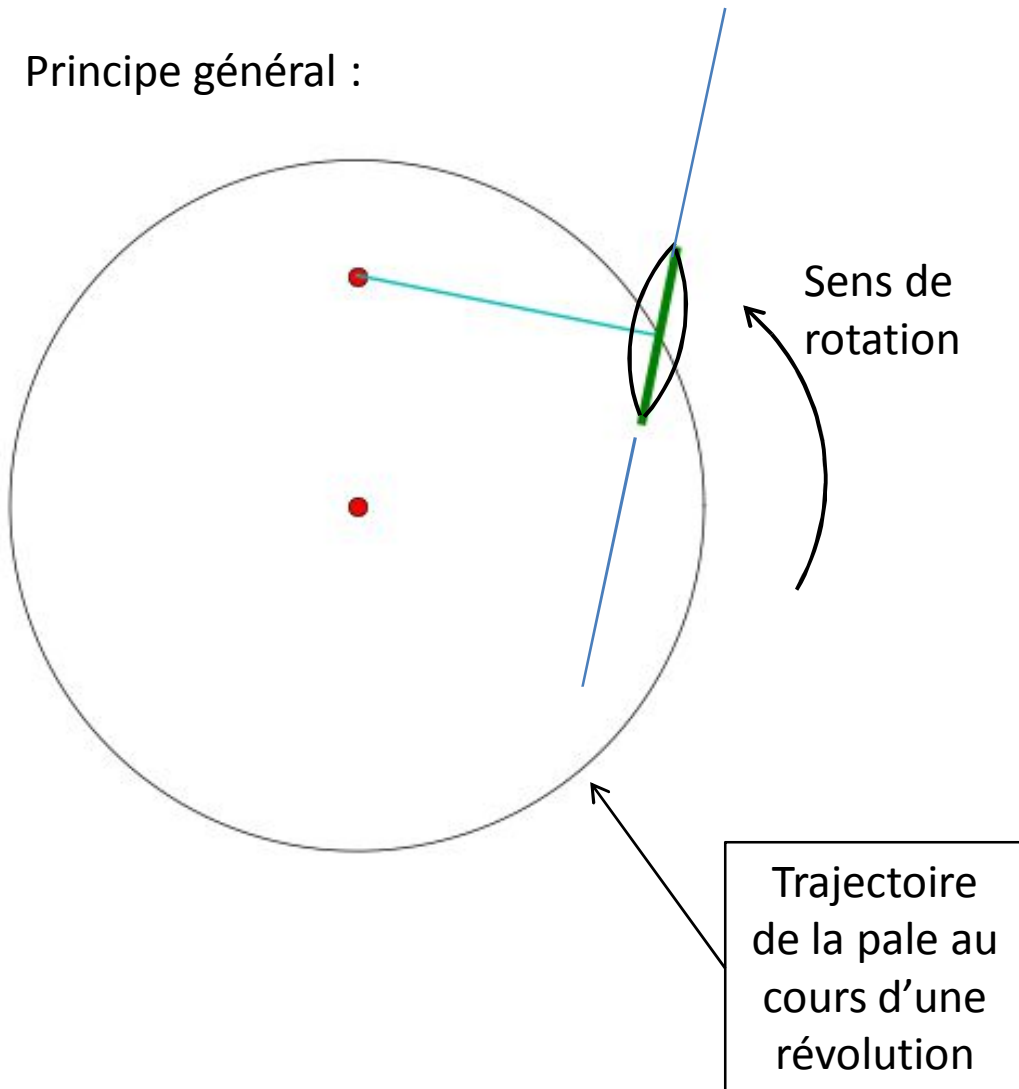
Quel est le principe de fonctionnement du VSP ?

- I - Modélisation de la propulsion d'une pale
- II - Élaboration des efforts fournis par les VSP au vu du modèle choisi
- III - Détermination des efforts s'appliquant sur notre maquette
- IV- Comparaison des résultats et conclusion



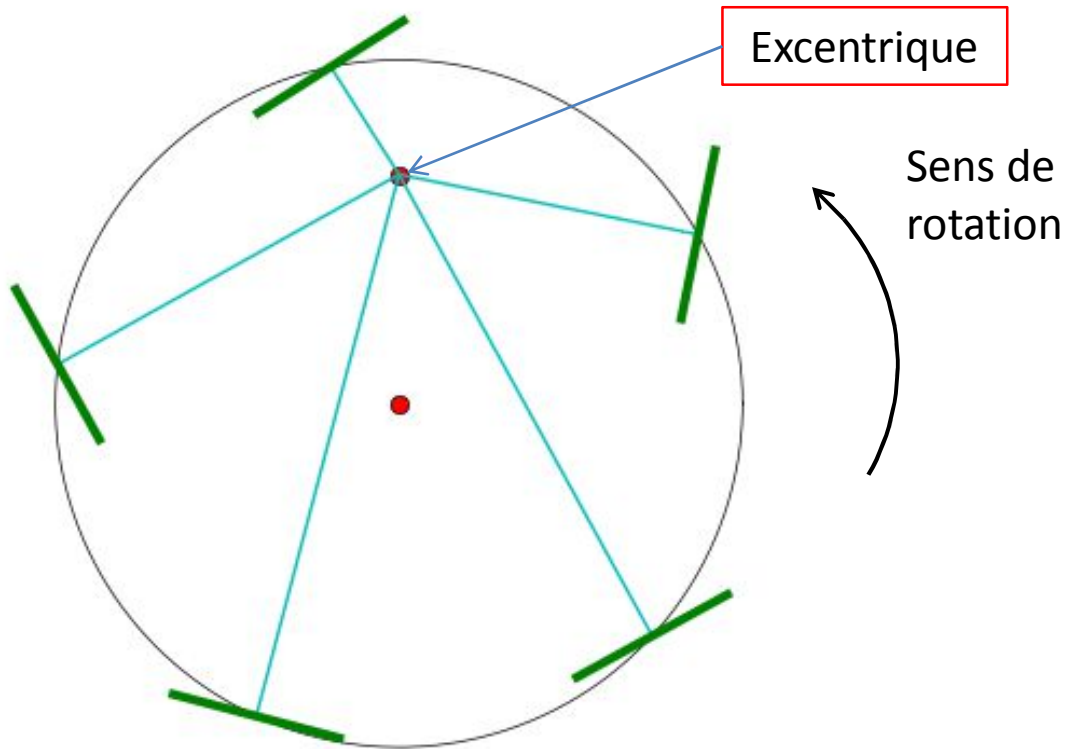
I- Modélisation du fonctionnement du VSP

Principe général :

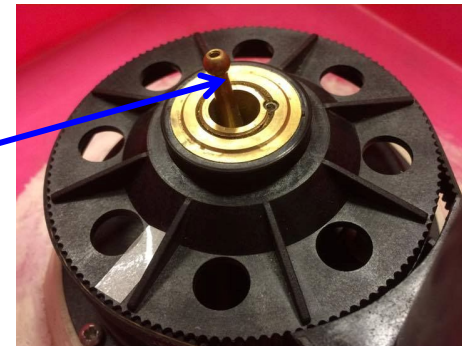


I- Modélisation du fonctionnement du VSP

Principe général : le rôle de l'excentrique



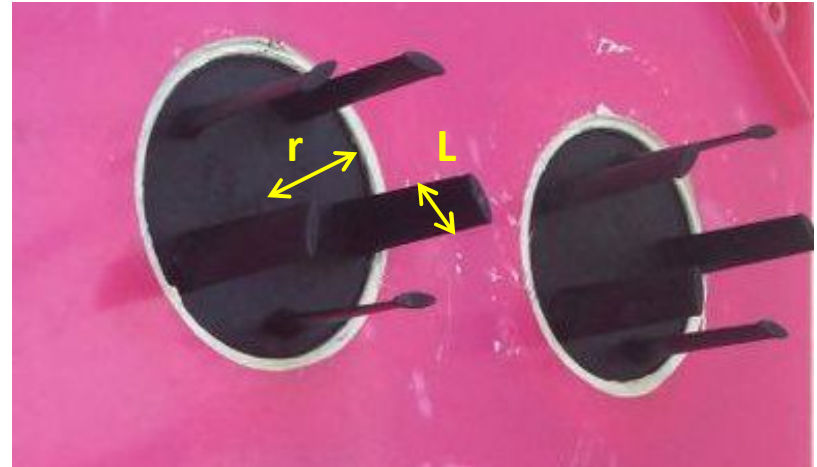
Dispositif de contrôle
de l'excentrique



I- Modélisation du fonctionnement du VSP

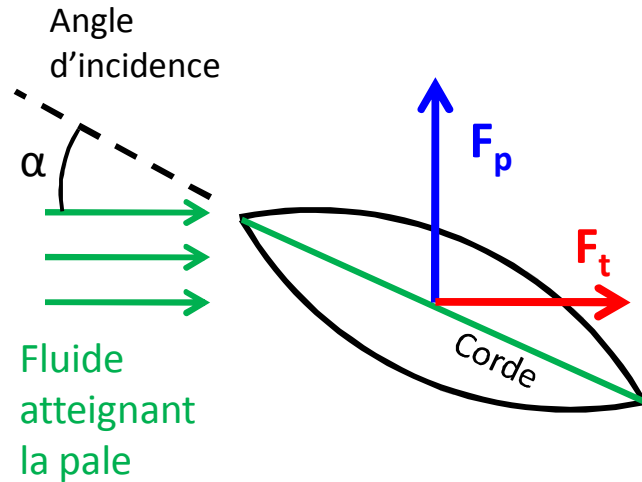
Nombre de Reynolds :

$$Re_{VSP} = \frac{\rho_{eau} \cdot \omega \cdot r \cdot L}{\eta_{eau}}$$



Modèle réduit : $\left\{ \begin{array}{l} r = 3 \text{ cm} \\ L = 1,5 \text{ cm} \\ \omega = 40 \text{ rad/s} \\ \eta_{eau} = 10^{-3} \text{ Pa}\cdot\text{s} \end{array} \right. \Rightarrow Re_{VSP} \approx 2 \cdot 10^4 \gg 2 \cdot 10^3 : \text{Écoulement turbulent}$

I- Modélisation du fonctionnement du VSP



Force de portance :

$$F_p = \frac{1}{2} \cdot \rho_{\text{fluide}} \cdot v^2 \cdot S \cdot C_p$$

Force de traînée :

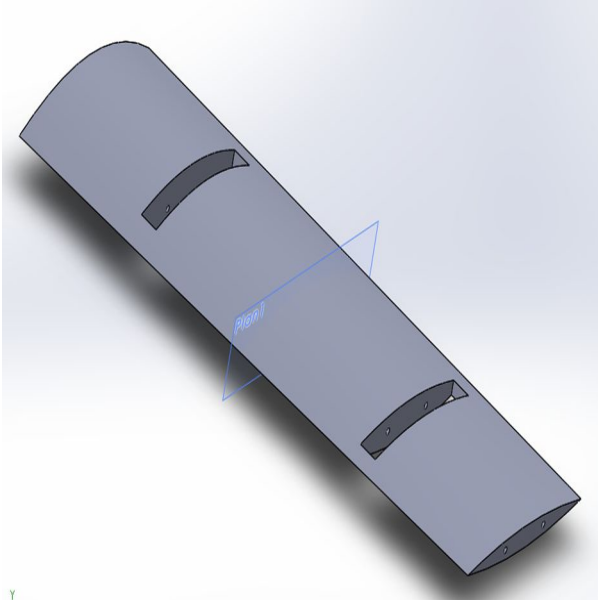
$$F_t = \frac{1}{2} \cdot \rho_{\text{fluide}} \cdot v^2 \cdot S \cdot C_t$$

Avec { ρ_{fluide} : masse volumique du fluide (en kg/m^3)
 v : vitesse relative du fluide (en m/s)
 S : surface alaire de la pale (en m)
 C_t : coefficient de traînée (sans dimension)
 C_p : coefficient de portance (sans dimension) }

Encore inconnus
à ce stade

II - Élaboration des efforts fournis par les VSP au vu du modèle choisi

- **Détermination des coefficients de portance et de traînée :
Étude en soufflerie**
 - Fabrication d'une pale par stratoconception



II - Élaboration des efforts fournis par les VSP au vu du modèle choisi



Vérification du **nombre de Reynolds**
dans le cadre de l'expérience :

$$Re_{pale} = \frac{\rho_{air} \cdot v \cdot L}{\eta_{air}}$$

$$\rho_{air} = 1,22 \text{ kg/m}^3$$

$$v = 10 \text{ m/s}$$

$$L = 10 \text{ cm}$$

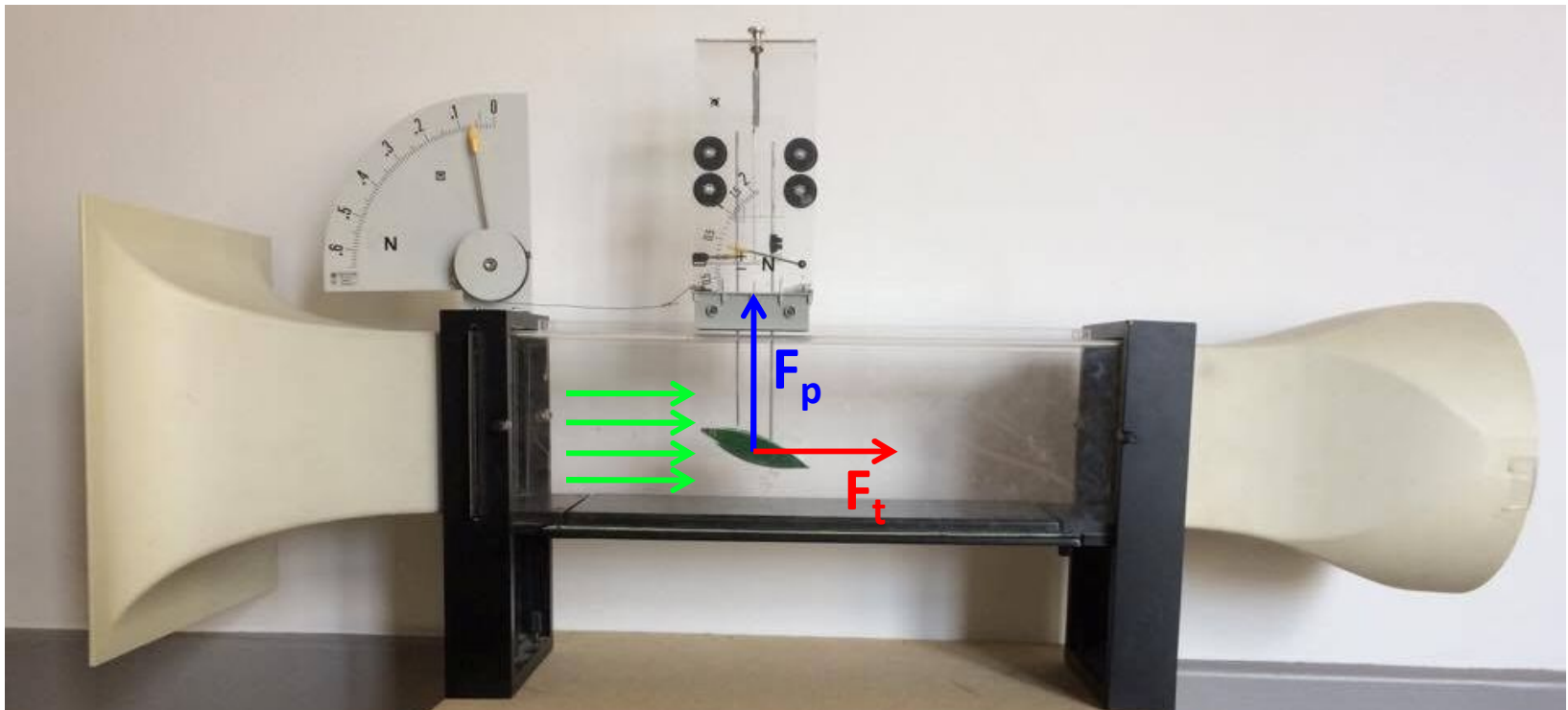
$$\eta_{air} = 10^{-5} \text{ Pa}\cdot\text{s}$$

$$\rightarrow Re_{pale} \approx 7 \cdot 10^4$$

$$\text{proche de } Re_{VSP} \approx 2 \cdot 10^4$$

II - Élaboration des efforts fournis par les VSP au vu du modèle choisi

- **Détermination des coefficients de portance et de traînée :
Étude en soufflerie**

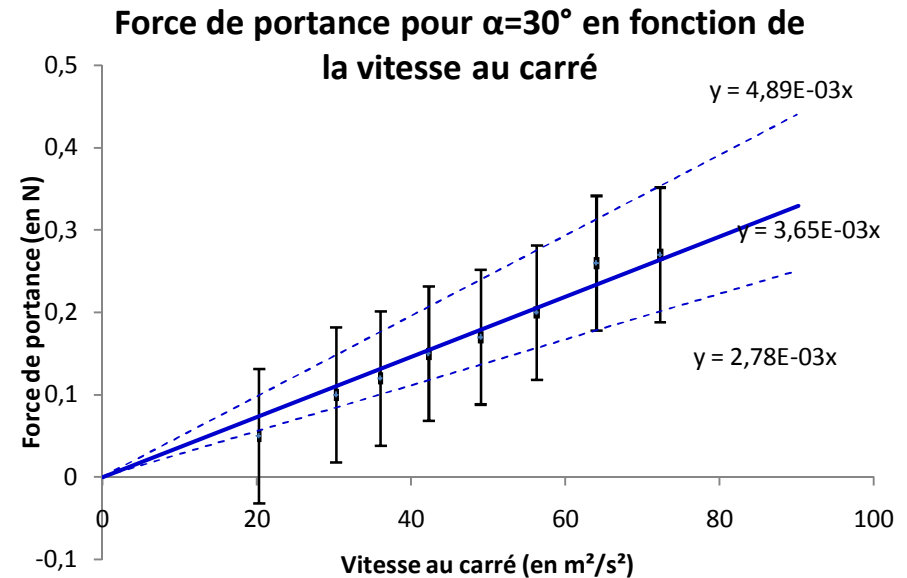
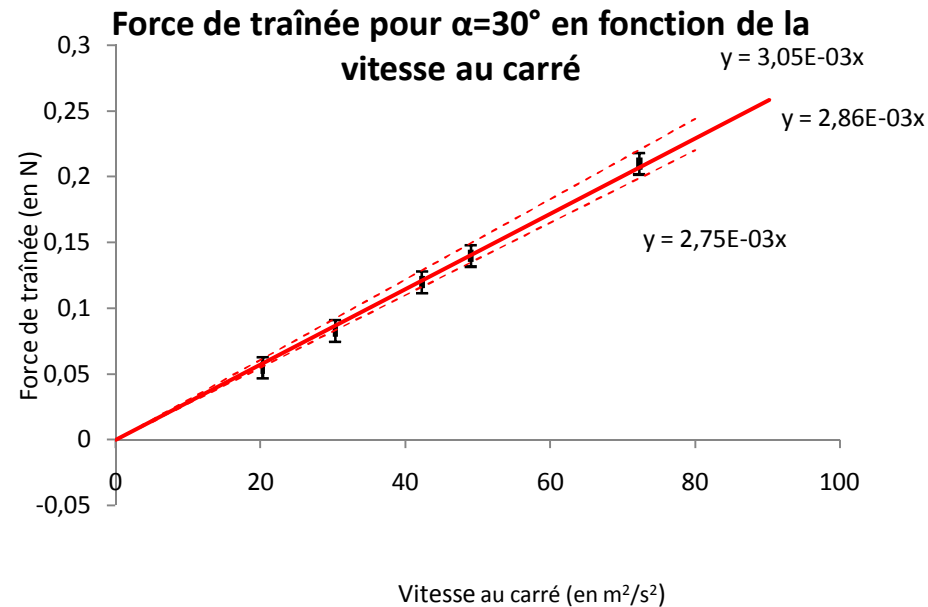


II - Élaboration des efforts fournis par les VSP au vu du modèle choisi

- Détermination des coefficients de portance et de traînée : Étude en soufflerie**

$$F_t = \frac{1}{2} \cdot \rho_{\text{fluide}} \cdot v^2 \cdot S \cdot C_t$$

$$F_p = \frac{1}{2} \cdot \rho_{\text{fluide}} \cdot v^2 \cdot S \cdot C_p$$



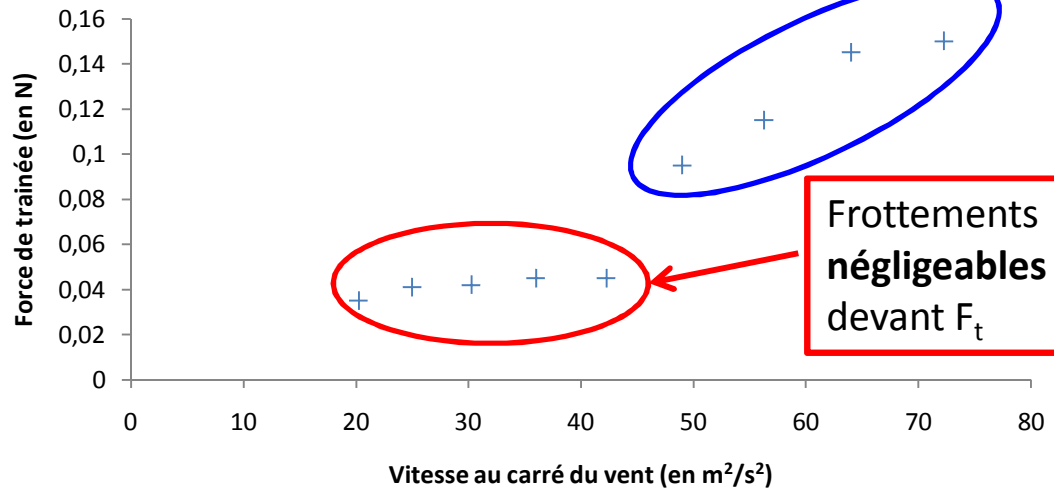
II - Élaboration des efforts fournis par les VSP au vu du modèle choisi

Analyse critique des résultats

Frottements dus au coulisement du dispositif sur les rails

Frottements **négligeables** devant F_t

Force de traînée en fonction de la vitesse au carré du fluide pour $\alpha=25^\circ$

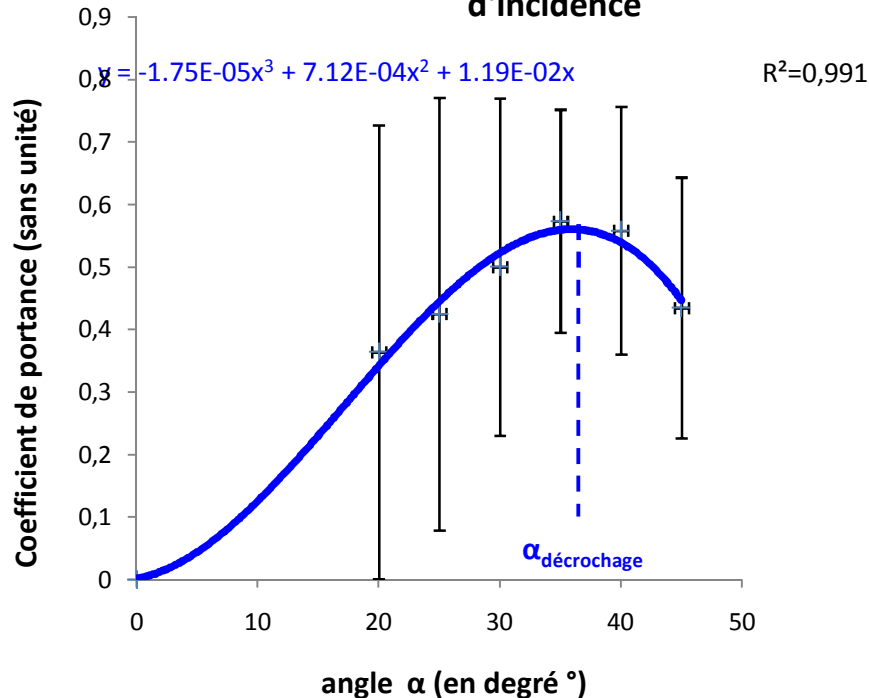


Frottements **non négligeables** devant F_t

II - Élaboration des efforts fournis par les VSP au vu du modèle choisi

- **Détermination des coefficients de portance et de trainée**

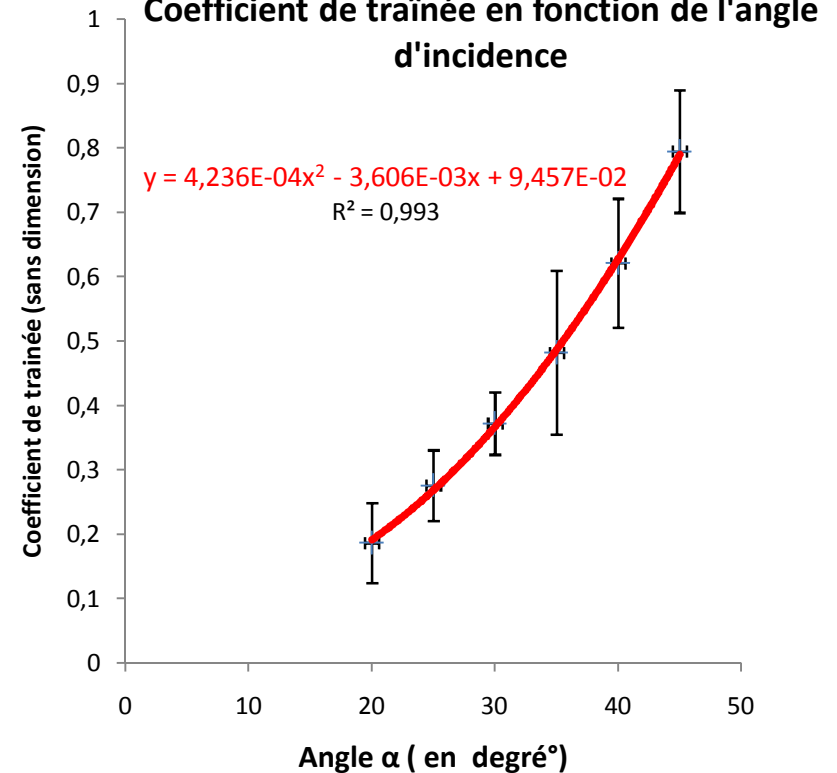
Coefficient de portance en fonction de l'angle d'incidence



Modélisation :

- Ordre 3 : phénomène de décrochage
- Profil symétrique de la pale : portance nulle pour $\alpha = 0^\circ$
- Coefficient de régression proche de 1

Coefficient de traînée en fonction de l'angle d'incidence

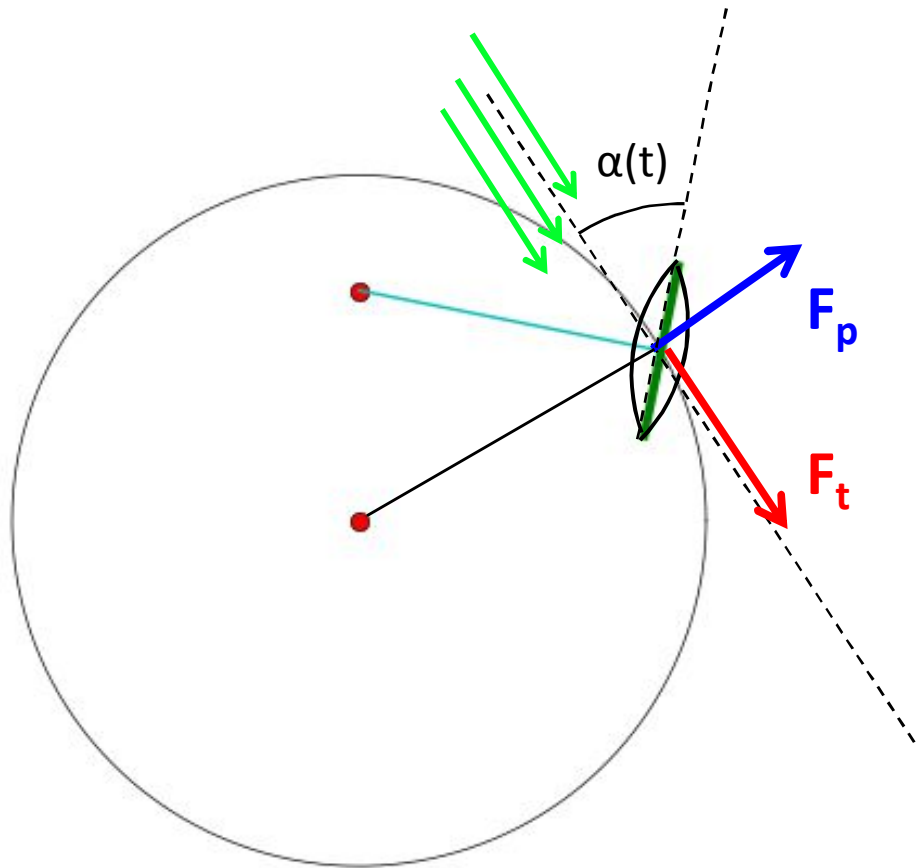


Modélisation :

- Ordre 2 avec coefficient de régression proche de 1
- pas de phénomène de décrochage : F_t est une fonction croissante de α

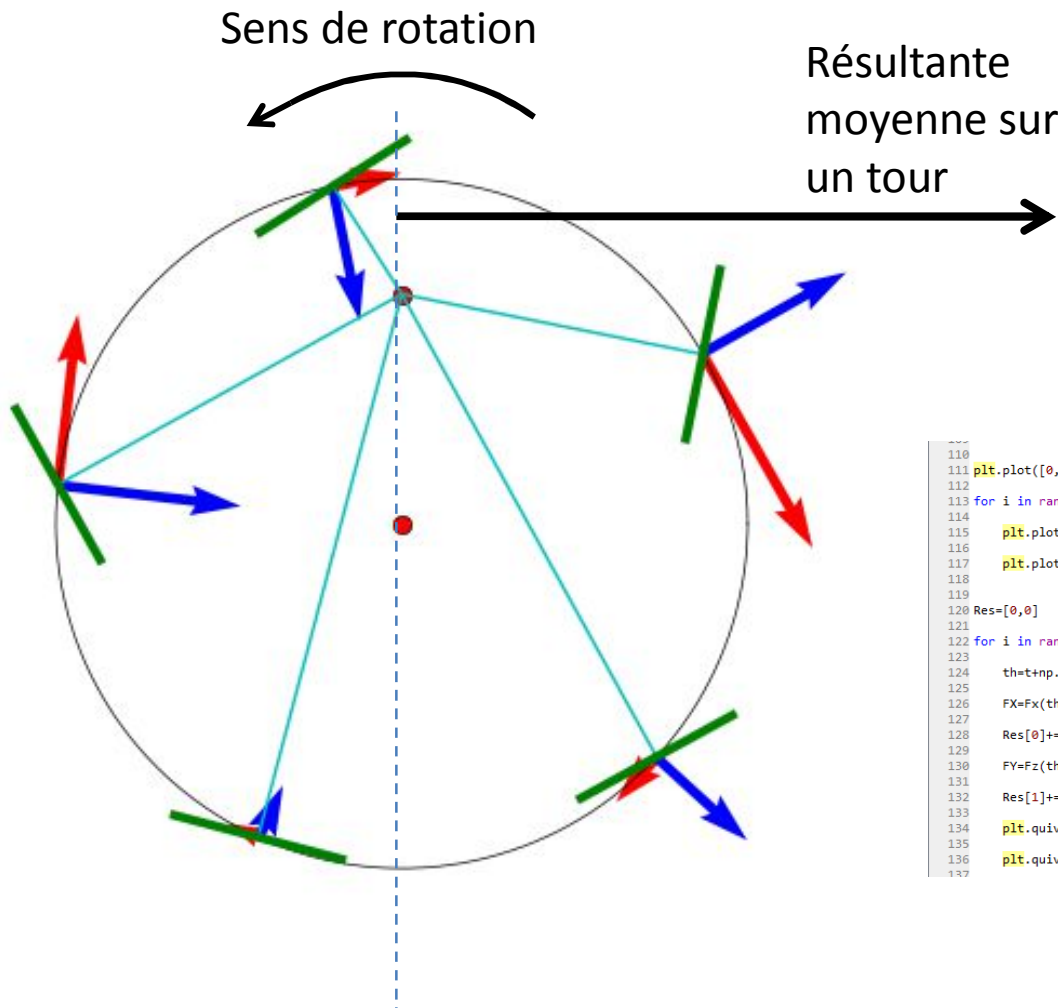
II - Élaboration des efforts fournis par les VSP au vu du modèle choisi

1^{er} cas : dessin des forces sur une pale



II - Élaboration des efforts fournis par les VSP au vu du modèle choisi

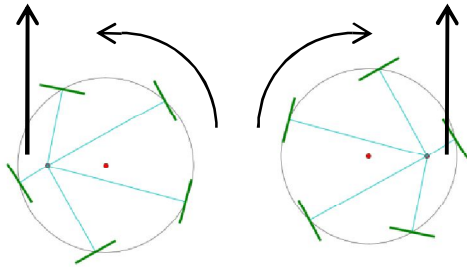
2^{ème} cas Représentation sur Python des efforts sur toutes les pales



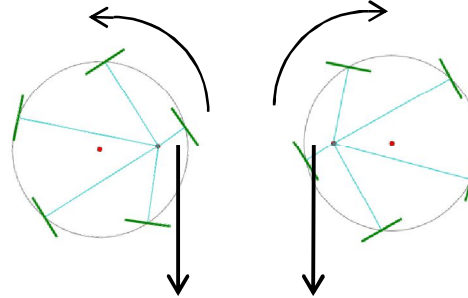
```
110
111 plt.plot([0,0],[0,e], 'or') #tracé de l'excentrique
112
113 for i in range(n): #tracé des rayons et des pales
114
115     plt.plot([0,r*np.cos(t+np.pi*2/n*i)],[e,np.sin(t+np.pi*2/n*i)*r], 'c')
116
117     plt.plot([P(t+np.pi*2/n*i)[0],Q(t+np.pi*2/n*i)[0]],[P(t+np.pi*2/n*i)[1],Q(t+np.pi*2/n*i)[1]], 'g', linewidth=1.0)
118
119
120 Res=[0,0] #résultante
121
122 for i in range(n): #tracé des vecteurs force
123
124     th=t+np.pi*2/n*i
125
126     FX=Fx(th)*np.sin(th)+Fz(th)*np.cos(th)
127
128     Res[0]+=FX
129
130     FY=Fz(th)*np.sin(th)-Fx(th)*np.cos(th)
131
132     Res[1]+=FY
133
134     plt.quiver(r*np.cos(th),r*np.sin(th),Fx(th)*np.sin(th),-Fx(th)*np.cos(th),angles='xy', scale_units='xy', scale=20)
135
136     plt.quiver(r*np.cos(th),r*np.sin(th),Fz(th)*np.cos(th),Fz(th)*np.sin(th),angles='xy', scale_units='xy', scale=20,
```

II - Élaboration des efforts fournis par les VSP au vu du modèle choisi

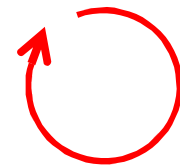
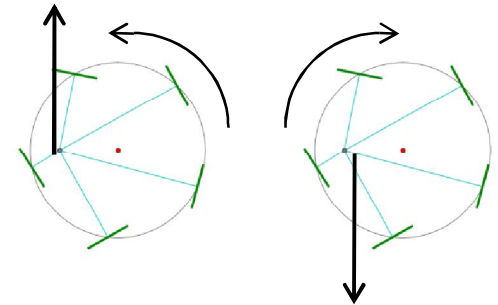
Pilotage du bateau équipé de VSP :



Marche avant



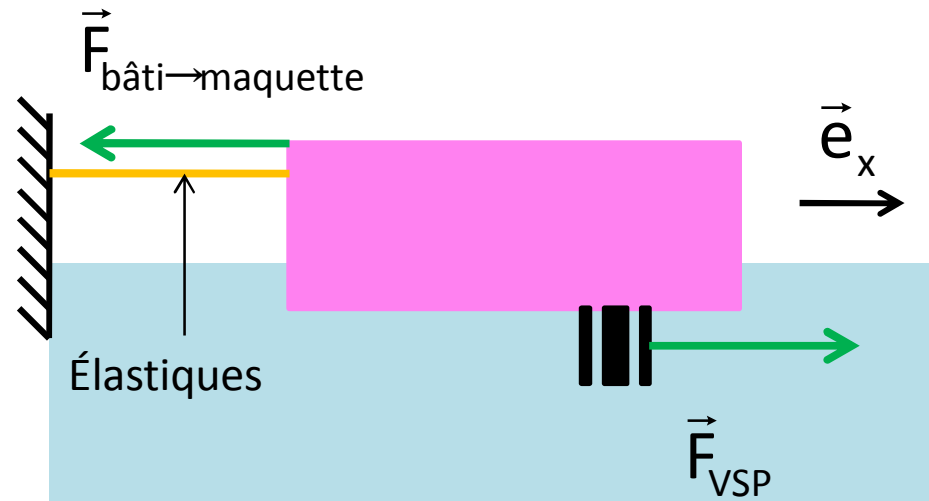
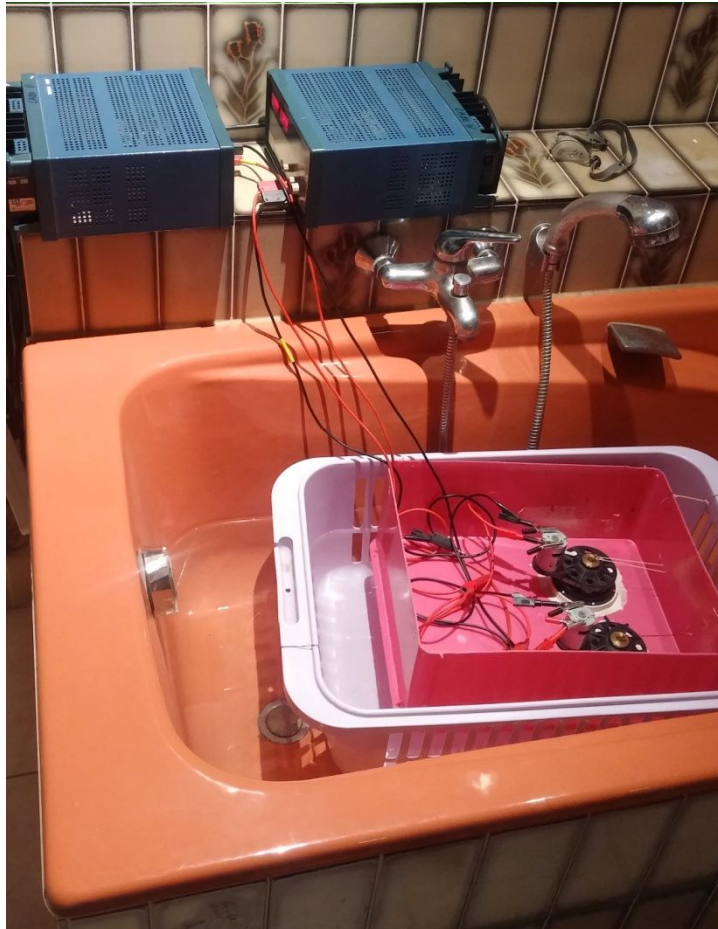
Marche arrière



Rotation

III - Détermination des efforts s'appliquant sur notre maquette

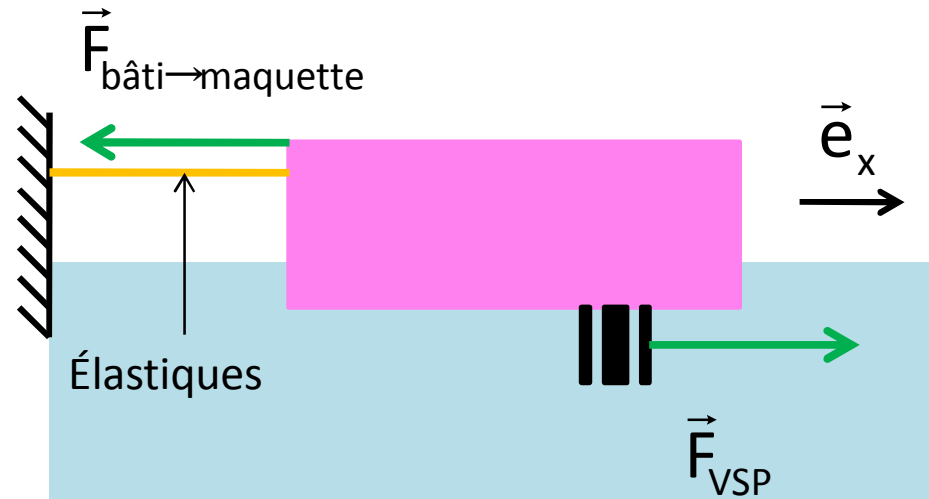
- Expérimentation de la maquette



III - Détermination des efforts s'appliquant sur notre maquette

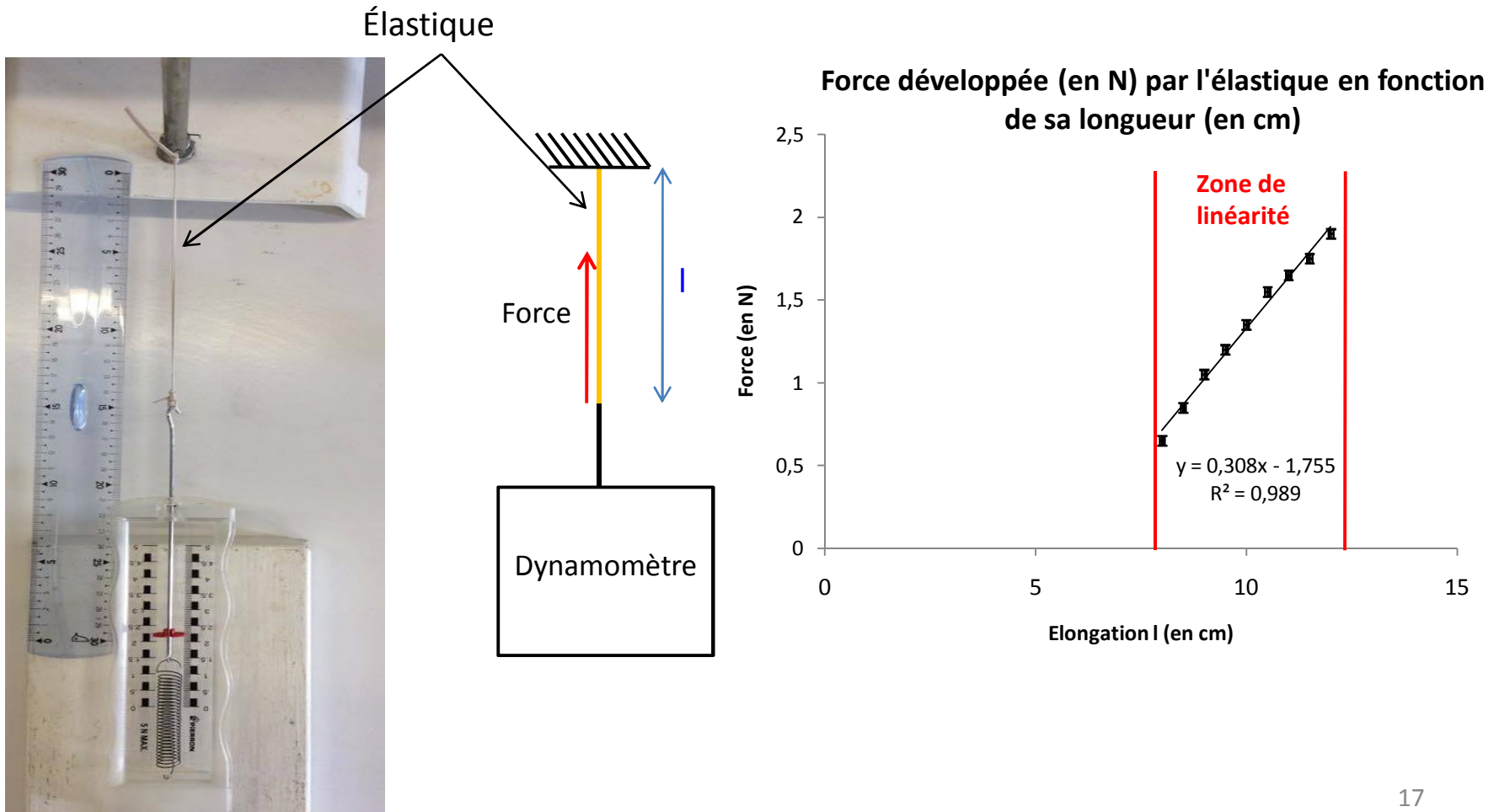
- **Hypothèses :**
 - On néglige l'actions des remous de l'eau sur la maquette devant les forces des élastiques et du VSP
 - Maquette statique lors de la photo
- **Théorème de la résultante statique** à notre maquette dans le référence terrestre supposé galiléen en projection dans sur \vec{e}_x :

$$F_{VSP} + F_{b\hat{a}ti \rightarrow maquette} + F_{eau \rightarrow maquette} = 0$$



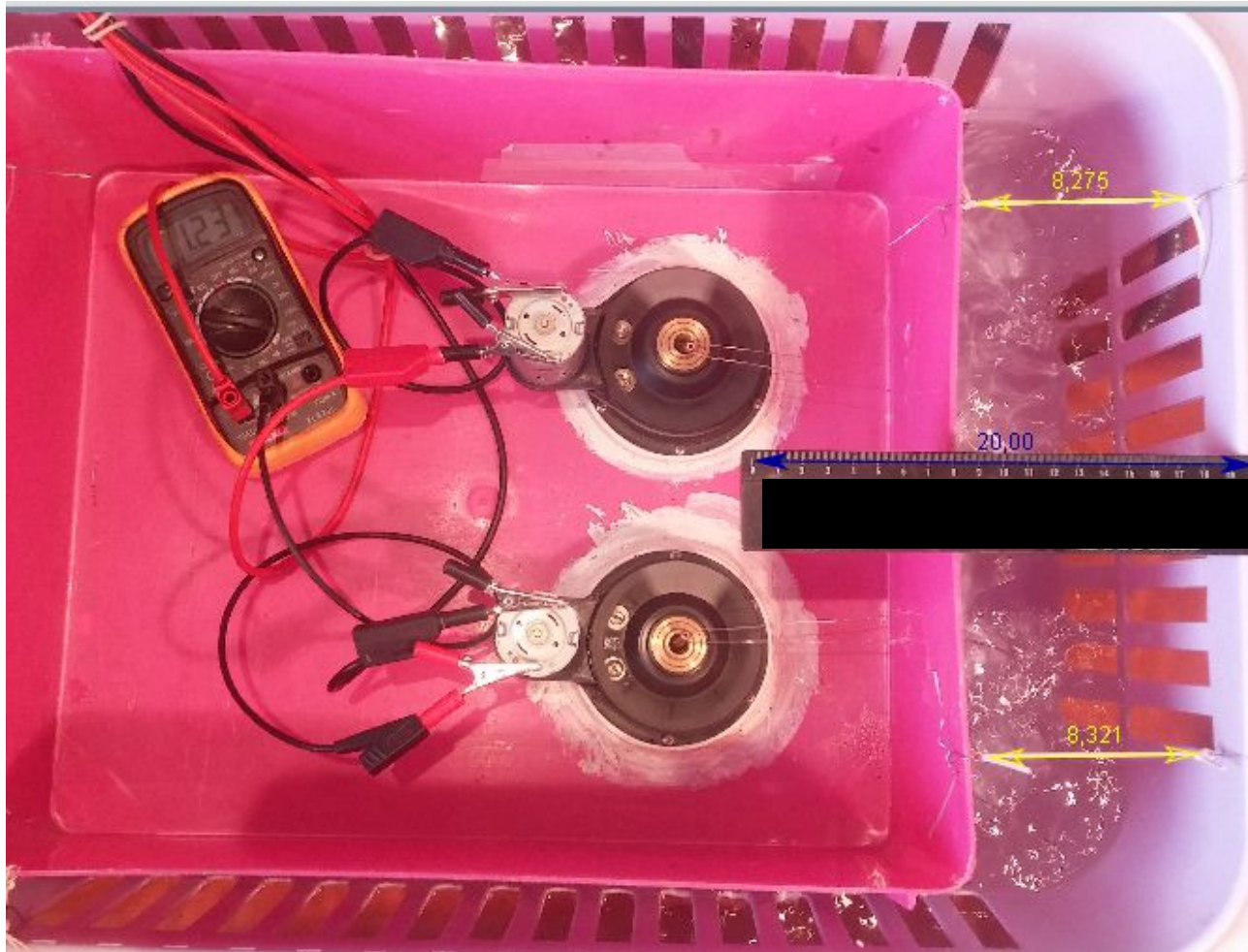
III - Détermination des efforts s'appliquant sur notre maquette

- Calcul de $F_{\text{bâti} \rightarrow \text{maquette}}$: détermination de la raideur des élastiques

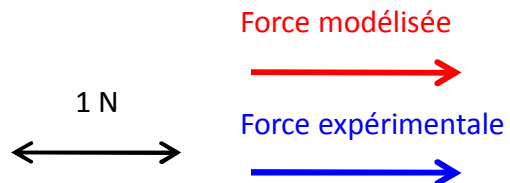
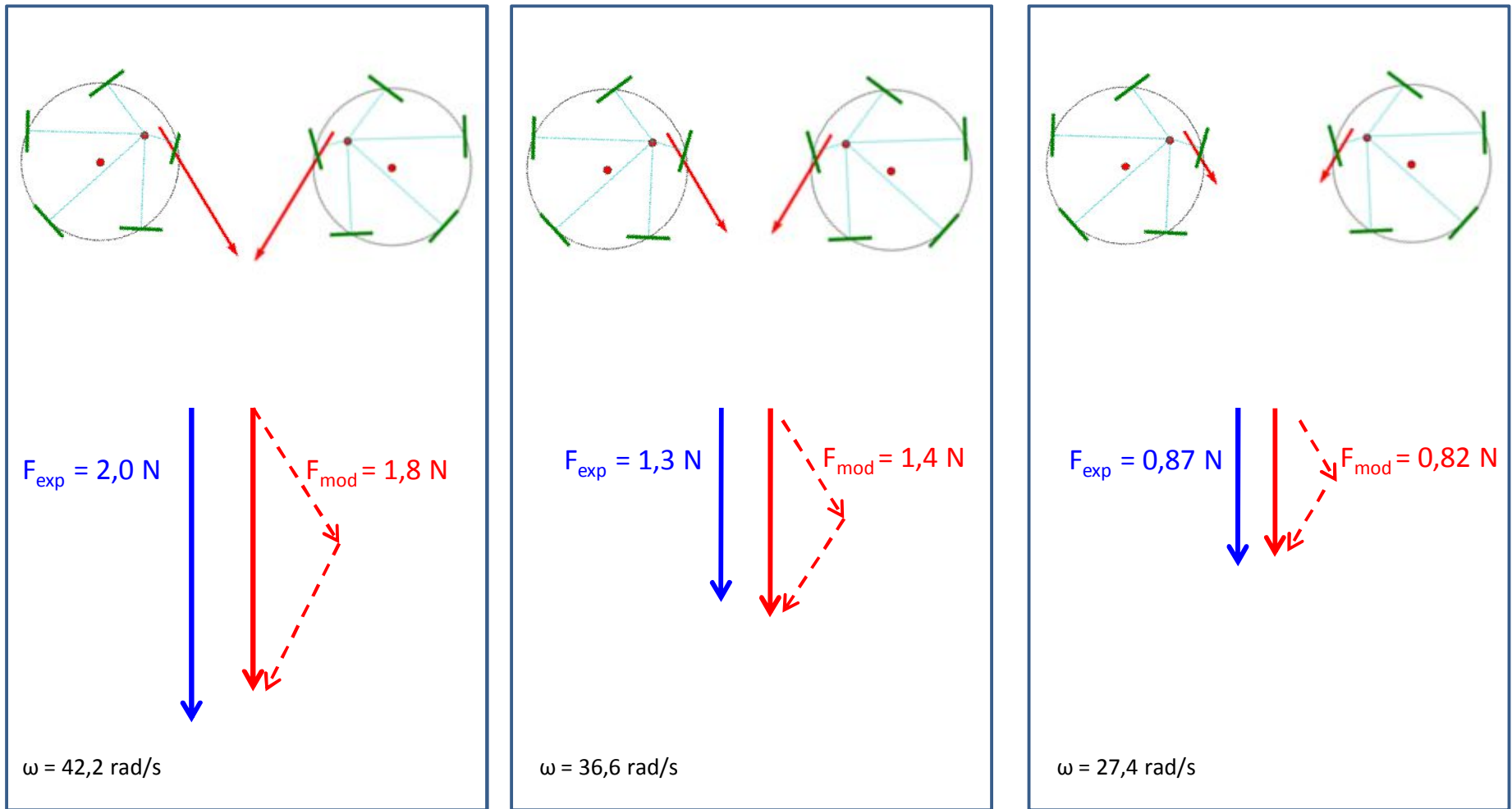


III - Détermination des efforts s'appliquant sur notre maquette

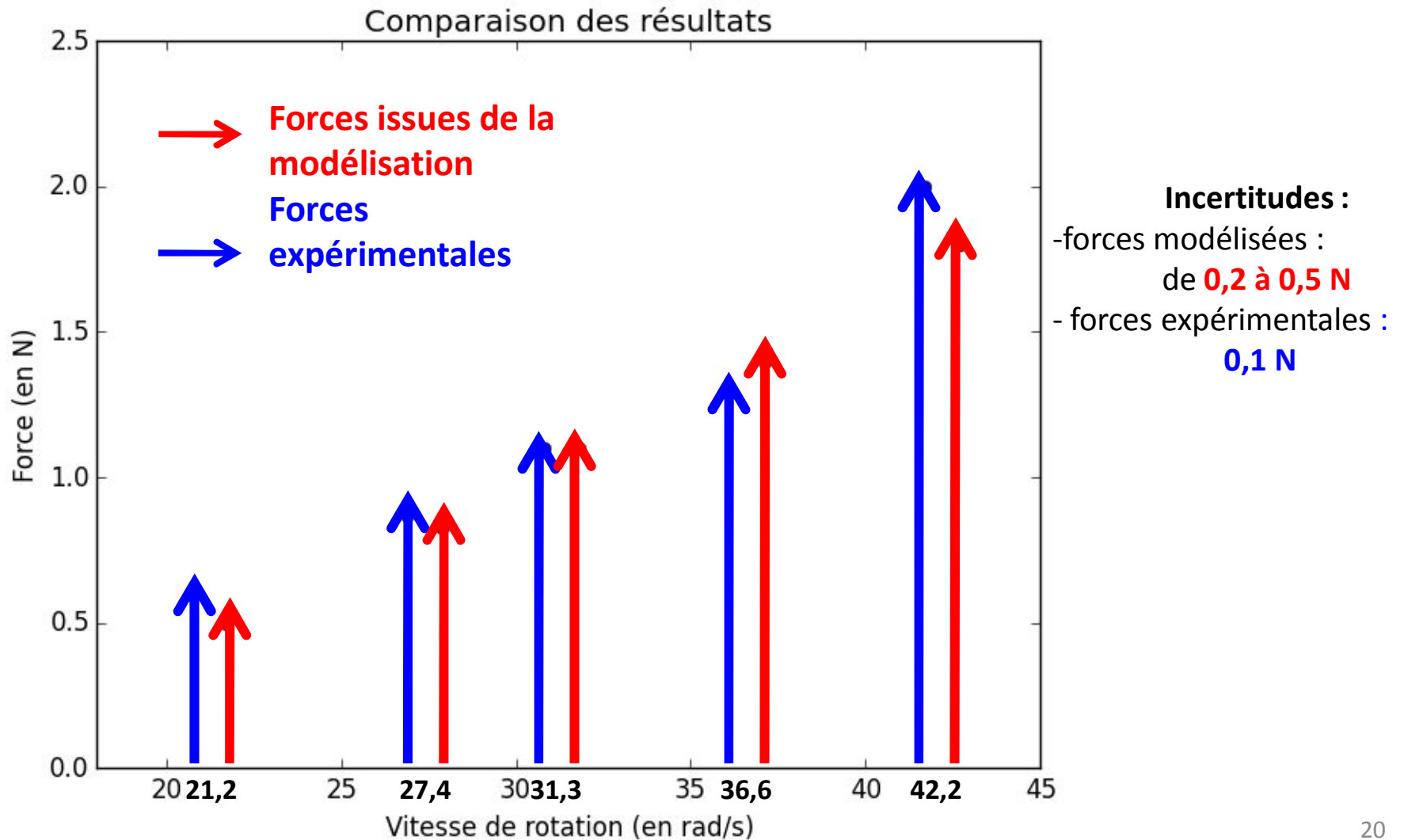
Exploitations et relevés des résultats :



IV- Comparaison des résultats et conclusion



IV- Comparaison des résultats et conclusion



Programme python :

```
1 import math
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 plt.clf()
6
7 n=5 #nombre de pales
8
9 t=np.pi/2 #instant initial
10
11 e=0.02 #excentrique
12
13 l=0.015/2 #largeur de la pale
14
15 r=0.03 #rayon du cercle
16
17 rho=1000 #masse volumique du fluide
18
19 S=0.015*0.05 #surface ailaire
20
21 w=42 #vitesse de rotation
22
23
24
25 def x(th): #distance excentrique-acroche de la pale
26     return math.sqrt(e**2+r**2-2*e*r*math.sin(th))
27
28
29
30 def gamma(th): #angle pale-tangente au cercle
31     resu=(2*r**2-2*e*r*np.sin(th))/(2*x(th)*r)
32
33     if resu>=1:
34         resu=1
35
36     elif resu<=-1:
37         resu=-1
38
39     return math.acos(resu)
40
41
42
43
44
45
46 def Cz(th): #coefficient de portance
47     a=gamma(th)*180/np.pi
48
49     return -1.75*10**(-5)*a**3+7.12*10**(-4)*a**2+1.19*10**(-2)*a
50
51
52
53 def Cx(th): #Coefficient de trainée
54     a=gamma(th)*180/np.pi
55
56     return 4.24*10**(-4)*a**2 - 3.61*10**(-3)*a + 9.46*10**(-2)
57
58
```

```

58
59
60 def P(th): #extrémité 1 de la pale
61
62     if 0<=th%(2*np.pi)<=np.pi/2 or 3*np.pi/2<=th%(np.pi*2)<=2*np.pi:
63
64         return [(r-l*np.sin(gamma(th)))*np.cos(th)+l*np.cos(gamma(th))*np.sin(th), (r-l*np.sin(gamma(th)))*np.sin(th)-l*np.cos(gamma(th))*np.cos(th)]
65     else:
66
67         return [(r+l*np.sin(gamma(th)))*np.cos(th)+l*np.cos(gamma(th))*np.sin(th), (r+l*np.sin(gamma(th)))*np.sin(th)-l*np.cos(gamma(th))*np.cos(th)]
68
69
70 def Q(th): #extrémité 2 de la pale
71
72
73     if 0<=th%(2*np.pi)<=np.pi/2 or 3*np.pi/2<=th%(np.pi*2)<=2*np.pi:
74
75         return [(r+l*np.sin(gamma(th)))*np.cos(th)-l*np.cos(gamma(th))*np.sin(th), (r+l*np.sin(gamma(th)))*np.sin(th)+l*np.cos(gamma(th))*np.cos(th)]
76     else:
77
78         return [(r-l*np.sin(gamma(th)))*np.cos(th)-l*np.cos(gamma(th))*np.sin(th), (r-l*np.sin(gamma(th)))*np.sin(th)+l*np.cos(gamma(th))*np.cos(th)]
79
80
81
82 def Fx(th): #Force de trainée
83
84     return 0.5*rho*S*(r*w)**2*Cx(th)
85
86
87 def Fz(th): #Force de portance
88
89
90     if 0<=th%(2*np.pi)<=np.pi/2 or 3*np.pi/2<=th%(np.pi*2)<=2*np.pi:
91
92         return 0.5*rho*S*(r*w)**2*Cz(th)
93     else:
94
95         return - 0.5*rho*S*(r*w)**2*Cz(th)
96
97
98
99 def MomentEn0(Fx,Fy,Mx,My,0x,0y):
100
101     return (Mx-0x)*Fy-(My-0y)*Fx
102
103
104 C=np.arange(0,2*np.pi,0.001) #tracé du cercle
105 X=[r*np.cos(c) for c in C]
106 Y=[r*np.sin(c) for c in C]
107
108 plt.plot(X,Y,'k')
109
110 plt.plot([0,0],[0,e], 'or') #tracé de l'excentrique
111
112 for i in range(n): #tracé des rayons et des pales
113
114     plt.plot([0,r*np.cos(t+np.pi*2/n*i)], [e,np.sin(t+np.pi*2/n*i)*r], 'c')
115

```



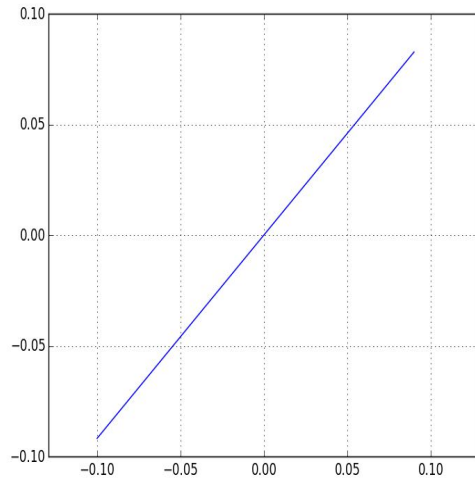
```

115
116 plt.plot([P(t+np.pi*2/n*i)[0],Q(t+np.pi*2/n*i)[0]], [P(t+np.pi*2/n*i)[1],Q(t+np.pi*2/n*i)[1]], 'g')
117
118
119 Res=[0,0] #résultante
120
121 for i in range(n): #tracé des vecteurs force
122
123     th=t+np.pi*2/n*i
124
125     FX=Fx(th)*np.sin(th)+Fz(th)*np.cos(th)
126
127     Res[0]+=FX
128
129     FY=Fz(th)*np.sin(th)-Fx(th)*np.cos(th)
130
131     Res[1]+=FY
132
133     plt.quiver(r*np.cos(th),r*np.sin(th),Fx(th)*np.sin(th),-Fx(th)*np.cos(th),angles='xy', scale_units='xy', scale=20,color='b')
134
135     plt.quiver(r*np.cos(th),r*np.sin(th),Fz(th)*np.cos(th),Fz(th)*np.sin(th),angles='xy', scale_units='xy', scale=20,color='b')
136
137
138
139 print('Résultante :',Res)
140
141 print('Norme résultante :',(Res[0]**2+Res[1]**2)**0.5)
142
143 plt.quiver(0,0.025,Res[0],Res[1],angles='xy',scale_units='xy', scale=20,color='r') #tracer résultante
144
145 plt.axis('Equal')
146 plt.axis([-0.08,0.08,-0.04,0.05])
147 plt.show()
148
149

```

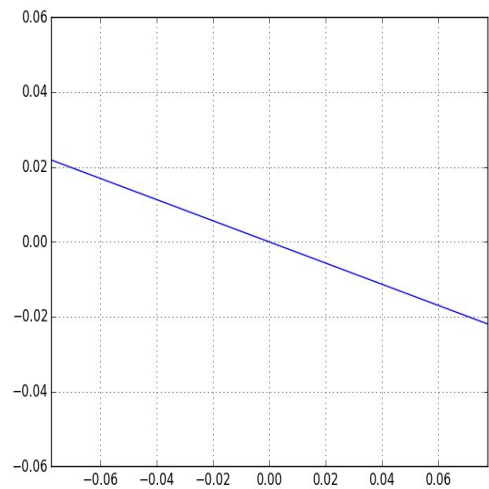
Annexe 1 : Détermination de la droite d'action de la résultante

Moment au point
 $(-0.0048, x+0.026)$
(en Nm)

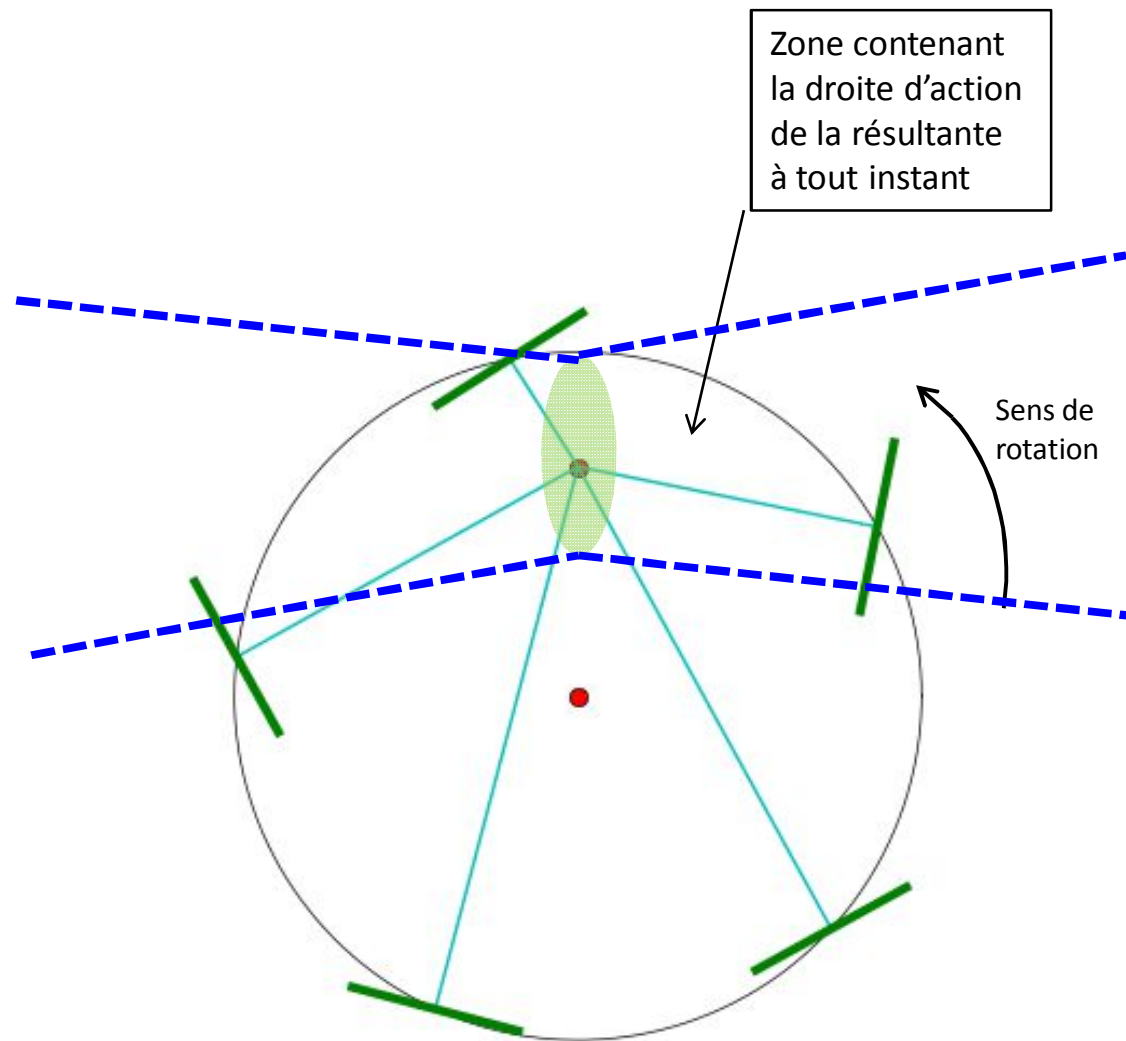


Distance au point $(-0.0048, 0.026)$ (en m)

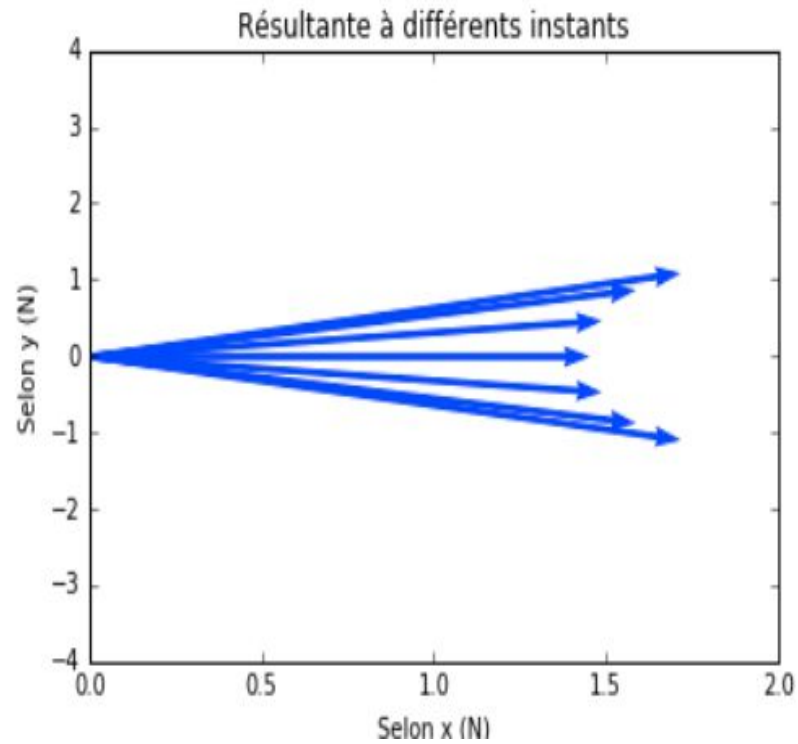
Moment au point
 $(x-0.0048, 0.026)$
(en Nm)



Distance au point $(-0.0048, 0.026)$ (en m)



Annexe 2 : Oscillation de la résultante au cours du temps



Annexe 3 : Influence du nombre de pales

