

**Un jeu olympique : le Kayak**

EL KARKOURI Mohamed Yahya

Juin 2024

**Table des matières :**

1-Introduction

2-Etude mécanique

3-Approche expérimentale

4-Critique et amélioration du modèle

5-Annexe

6-Bibliographie

**1-Introduction :**

Le kayak est un sport maritime très prisé de nos jours et qui sera présent lors des JO PARIS 2024. C'est principalement une course sur eau douce, c'est pour cela que l'augmentation de la vitesse des kayakistes est l'enjeu central autour de ce sport. L'étude de ce sport olympique nous aidera principalement à trouver des moyens d'améliorer les performances sportives des athlètes.

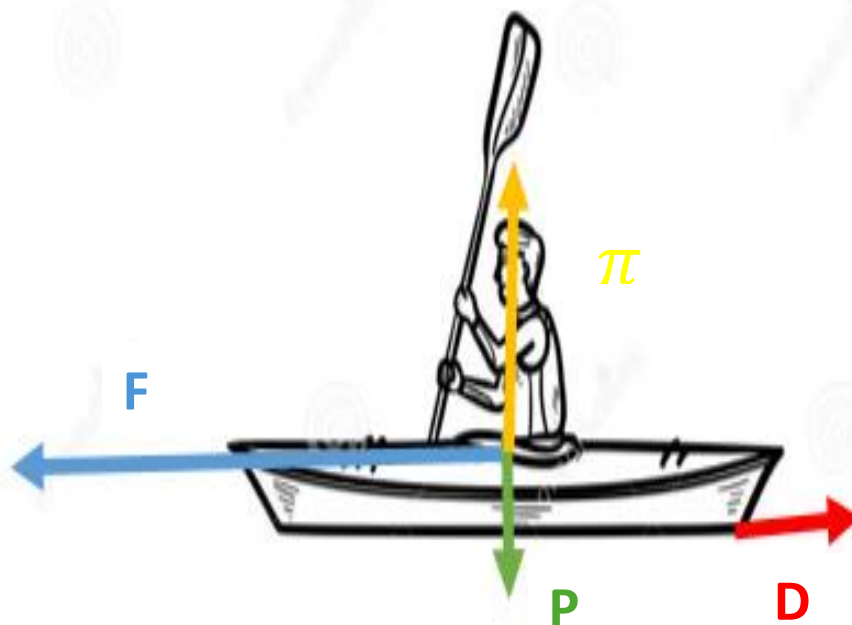
**2 -Etude mécanique :****2-1 Forces auxquelles est soumis le kayakiste :**

Fig 1

Notre système d'étude : {le kayak + l'athlète + la pagaie}

Bilan des actions mécaniques :

P : poids du système

$\pi$  : la poussée d'Archimède

D : la force de trainée (aérodynamique + hydrodynamique)

F : la force de propulsion

*NB :*

Étant donné que le poids est entièrement compensé par la poussée d'Archimède (tous deux portés horizontalement), et vu que la force de propulsion est inhérente aux capacités physiques du kayakiste et de plusieurs paramètres difficilement cernables, nous nous intéresserons par la suite seulement à la force de trainée.

La force de trainée a 2 composantes :

Une composante aérodynamique : qui d'après l'hypothèse de Froude est subdivisée en deux actions :

- Une force de frottement pure :

$$D_f = \frac{1}{2} \rho V_H^2 A_w C_f$$

- Action des vagues sur la coque :

$$D_w = \frac{1}{2} \rho V_H^2 C_w \frac{\Delta^{5/3}}{L^3}$$

Où :

- $C_f$  : coefficient de frottement pure qui dépend principalement du nombre de Reynolds (1) :

$$Re = V_H L / \nu$$

- $C_w$  : coefficient des vagues qui dépend essentiellement du nombre de Froude (2) :

$$Fr = V_H / \sqrt{gL}$$

- $A_w$  : surface à l'eau qu'on approchera par l'expression :

$$A_w = 2,5\sqrt{\Delta L}$$

*NB :*

Précédemment et dans toute la suite du rapport :

- $V_H$  : La vitesse de notre système
- $L$  : Longueur du kayak
- $\Delta$  : Volume du système qu'on approche à celui du kayak
- $\nu$  : viscosité de l'eau

Une composante aérodynamique qu'on écrira :

$$D_H = \frac{1}{2} \rho V_H^2 A_A$$

Où  $A_A$  : est l'aire frontale aérien

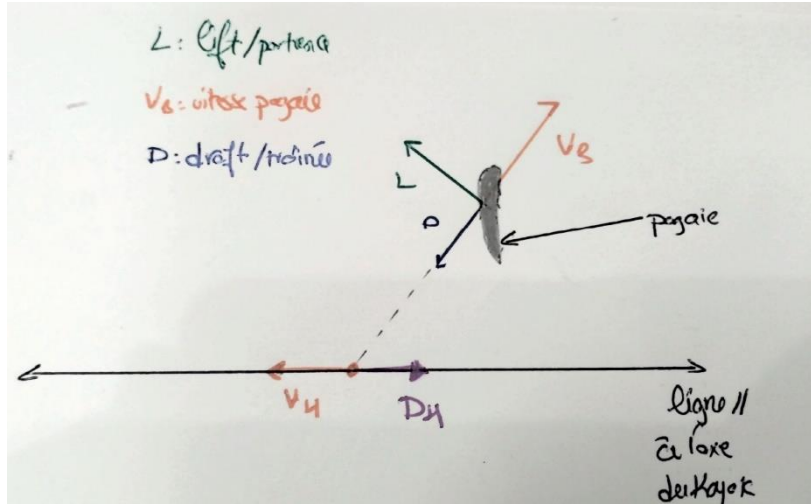
De là et à partir des deux composantes vues précédemment on pourra définir une force de trainée totale de la forme :

$$D_H = \frac{1}{2} \rho V_H^2 A_H (1)$$

Où  $A_H$  est un coefficient total dépendant des paramètres définis auparavant

**2-2 Efficacité et puissance :**

Fig 2 :



Étant donné que la pagaie a un mouvement périodique responsable de la mobilité du kayak, la force de trainée s’opposant au mouvement kayak est :

$$D_H = \frac{1}{T} \int_0^T (D \cos \theta + L \sin \theta) dt$$

Où

- $\theta$ : l’angle entre la ligne d’axe du kayak et  $V_B$
- $T$ : la période de pagayage

La puissance  $P$  à fournir par le kayakiste devra en partie compenser la trainée de la pagaie dans l’eau et « fera en sorte de » propulser le kayak, autrement dit :

$$P = D_H V_H + \frac{1}{T} \int_0^T D V_B dt$$

Le terme «  $D_H V_H$  » est le terme de puissance utile (fait avancer le kayak), il est donc légitime de définir un rendement  $\eta$  :

$$\eta = D_H V_H / P (2)$$

Comme il nous est impossible de cerner tous les paramètres cinétiques (vitesse de la pagaie dans l’eau...), dynamiques (forces de trainée...) et d’autres variables et difficilement quantifiable, mesurables, ou même prédictibles (le mouvement et l’angle d’entrée de la pagaie dans l’eau, sa forme et son matériau de construction...) dont dépend le rendement  $\eta$ , nous nous contenterons de limiter l’étude là-dessus. Par ailleurs on a récemment essayé d’exprimer ce rendement-là à travers les différents paramètres cités précédemment :

$$\frac{1}{\eta} = 1 + \frac{k}{\cos\phi} \frac{A_H}{A_V} \frac{T V_H}{\sqrt{A_V}}$$

Ce qui justifie tout de même la difficulté d'approfondir l'étude de cette notion.

De (1) et (2) on aboutit à :

$$P = \frac{1}{2} \frac{\rho A_H V_H^3}{\eta} \quad (3)$$

## 2-Approche expérimentale :

La relation simple (3) est un point clé à ce stade car relie directement la vitesse, le paramètre cinétique essentiel qui nous intéresse réellement car augmenter la vitesse équivaut à améliorer les performances sportives des kayakistes, la puissance à fournir par le kayakiste, le rendement  $\eta$  et les paramètres géométriques de la coque ( $A_H$ ).

La puissance et le rendement étant comme affirmé auparavant difficilement cernables et dépendent bien plus que des propriétés physiques et géométriques de la coque, l'approche expérimentale se restreindra à vérifier l'influence de la force de traînée (et donc du coefficient de frottement) sur la vitesse du kayak et retrouver expérimentalement la valeur du coefficient de frottement.

### 2 -1 Principe de l'expérience :

Nous disposons de 2 modèles miniatures de kayak Alpha de masse  $M_1$  et Oméga de masse  $M_2$  (dont on donnera la description en section 2 2)

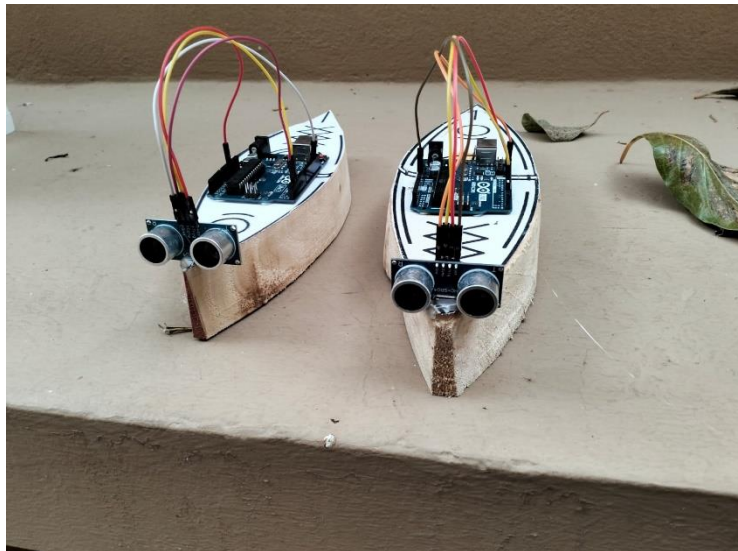


Fig 3 :

On les plonge dans un bassin d'eau d'environ 1 mètre de long. On leur donne un élan au début de l'expérience et on mesure leur vitesse au cours du temps chacun à son tour.

Par application du PFD :

$$M_e \frac{dV}{dt} = F_m(t) - F_D(t)$$

Où :

- $M_e$ : masse du prototype
- $V$ : vitesse du prototype

- $F_m(t)$  : la force de propulsion(nulle à tout instant car l'élan est donné au début seulement)
- $F_D(t)$ :la force de trainée qui d'après la section (1) s'écrit :  

$$\frac{1}{2}\rho S C_D V^2$$
- $S C_D$ :coefficient de frottement

Donc : 
$$M_e \frac{dV}{dt} = -\frac{1}{2}\rho S C_D V^2$$

Ainsi : 
$$\frac{V_0}{V(t)} = 1 + \frac{t}{\tau} \quad \text{où} \quad \tau = \frac{2M_e}{\rho S C_D V_0}$$

On devra par la suite s'attendre à une courbe de vitesse :

Fig 4 :

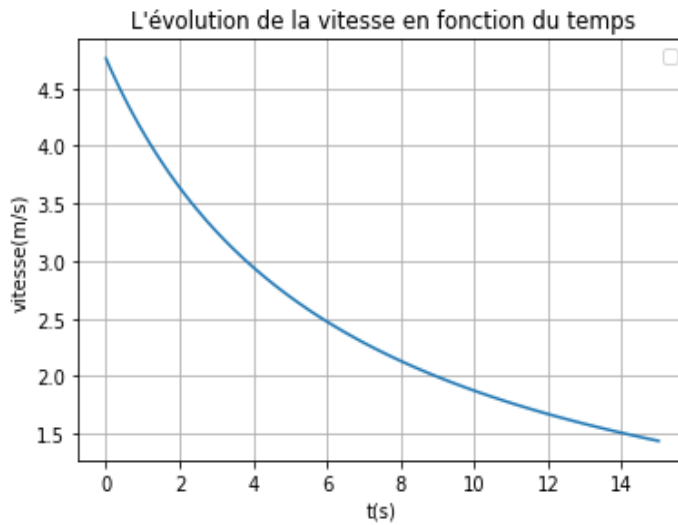
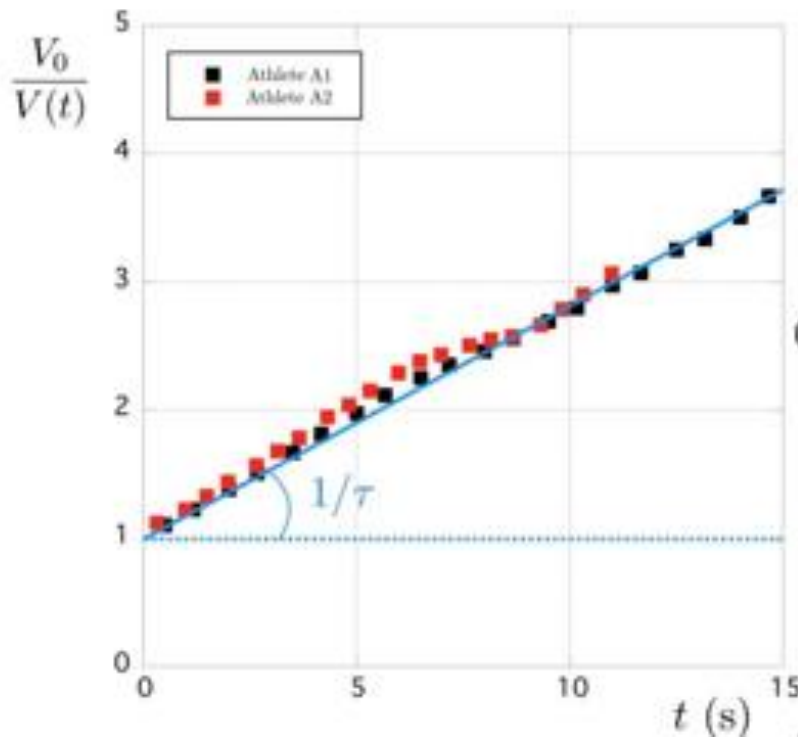


Fig 5 :

(source :On The physics Of Kayaking)



**2-2 Réalisation et mise en place de l'expérience :**



Fig 6

Les 2 modèles Alpha et Gamma sont fabriqués manuellement en bois, ayant tous 2 une forme géométrique proche des formes réelles en compétition mais qui diffèrent en masse et en aire frontale, dont on teste la flottabilité :

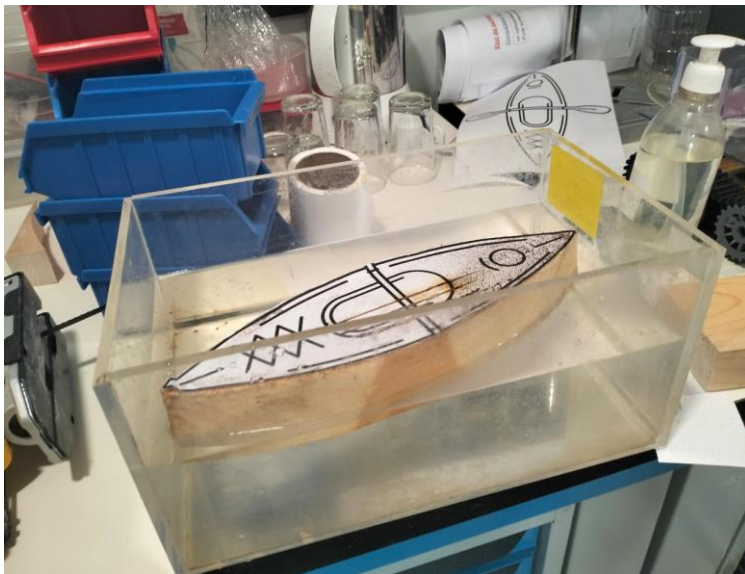


Fig 7

On équipe ensuite les 2 modèles chacun d'un capteur ultrason HC-SR04 et d'une carte d'acquisition Arduino qu'on étalonne au préalable:



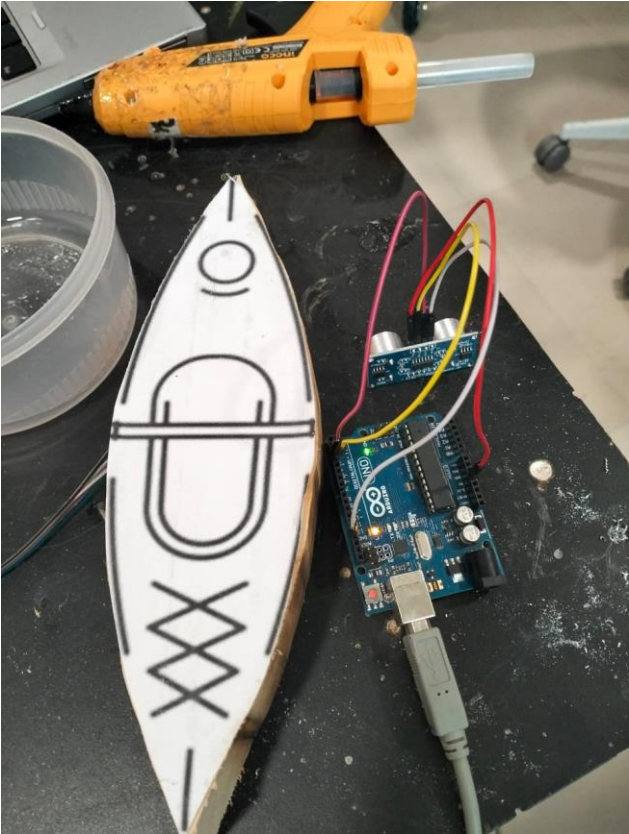


Fig 8

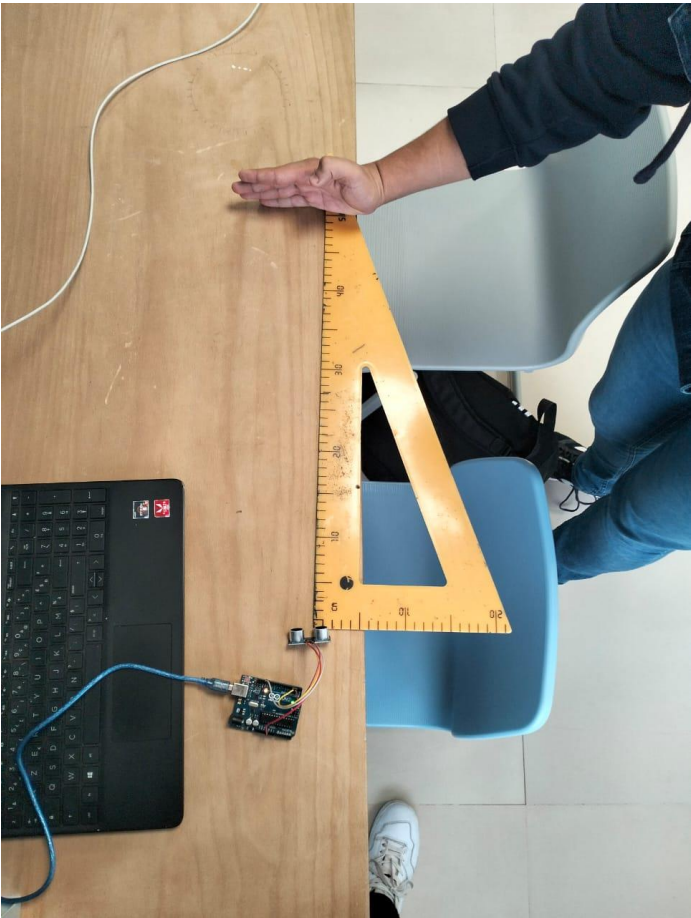


Fig 9

Une fois le modèle dans l'eau, la carte d'acquisition Arduino est relié par un câble à un ordinateur qui donne en temps réel l'évolution de l'abscisse.



Fig 10

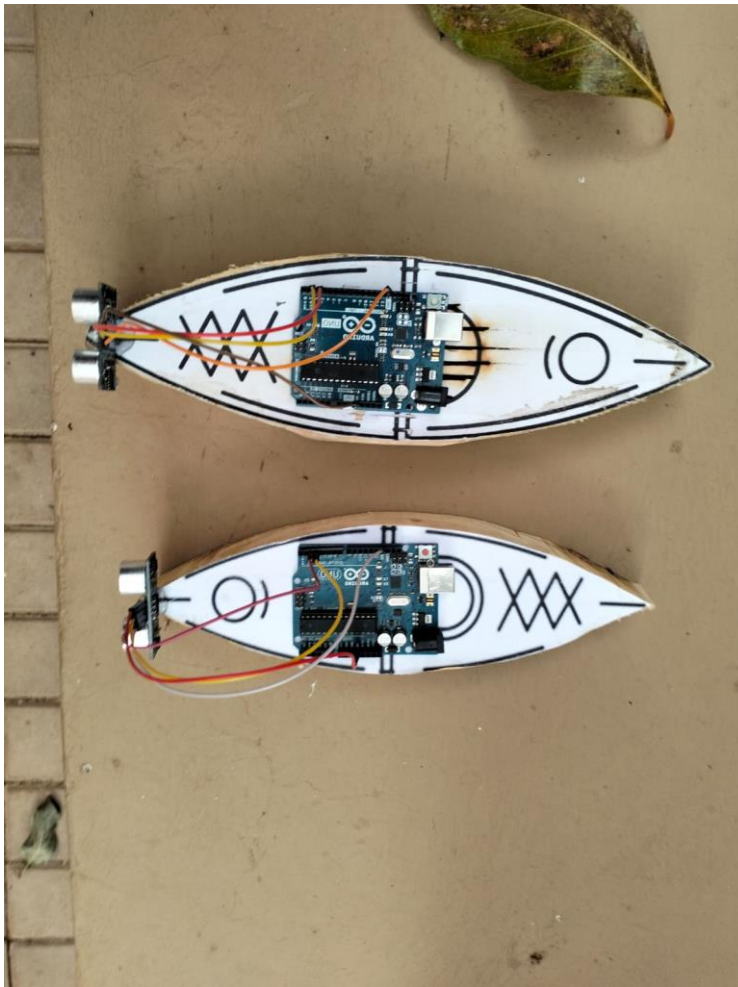


Fig 11



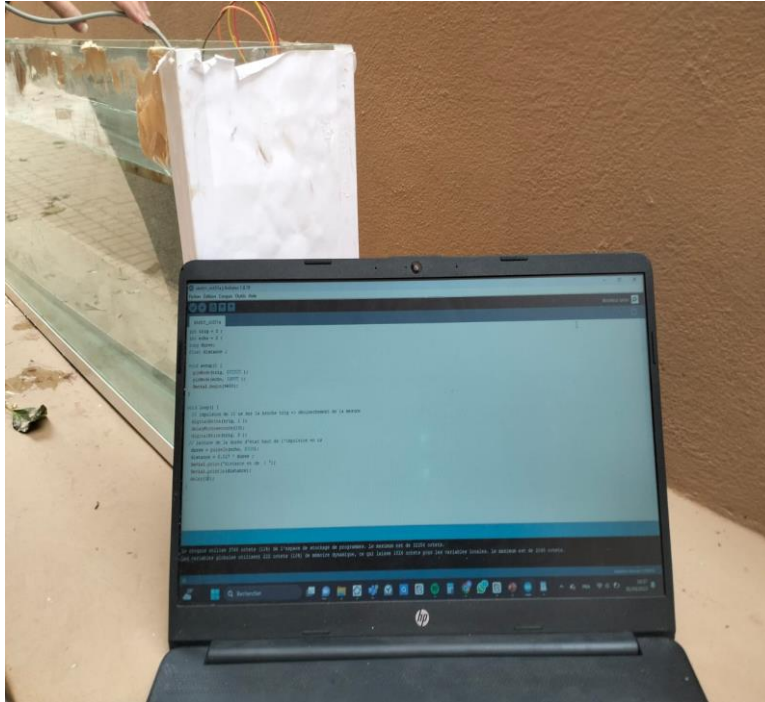


Fig 12

**2-3 Résultats obtenus et traitement informatique :**

On regroupe ensuite dans 2 listes python les données de temps et d'abscisse et on trace l'évolution de l'abscisse en fonction du temps (le « 0 » est choisi arbitrairement et l'abscisse est celui de la pointe du kayak) :

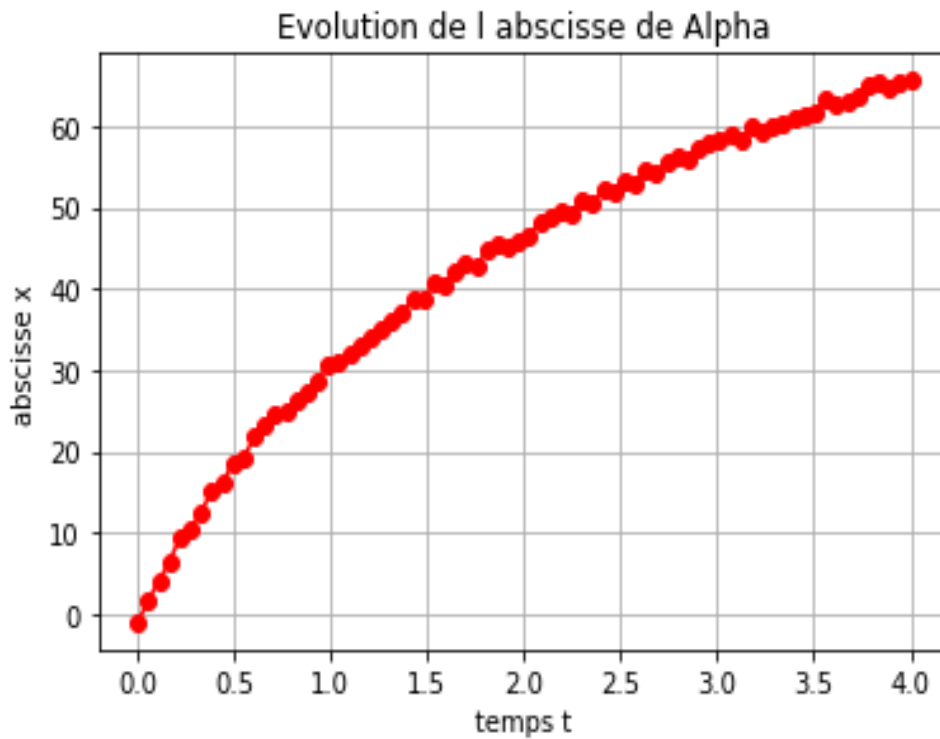


Fig 13

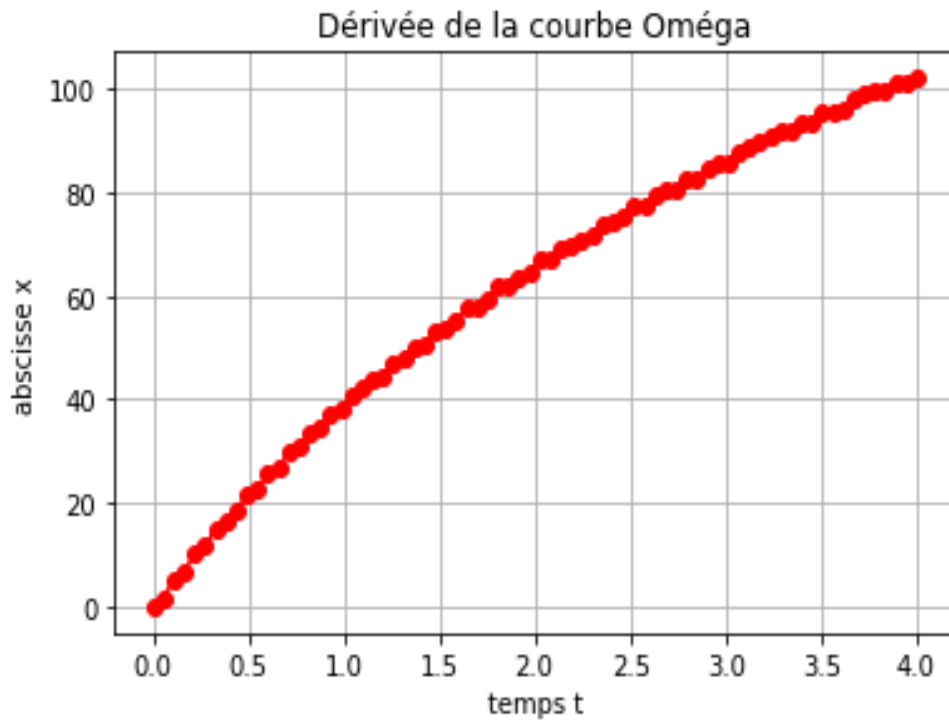


Fig 14

Quant aux courbes de vitesse on les obtient numériquement par python :

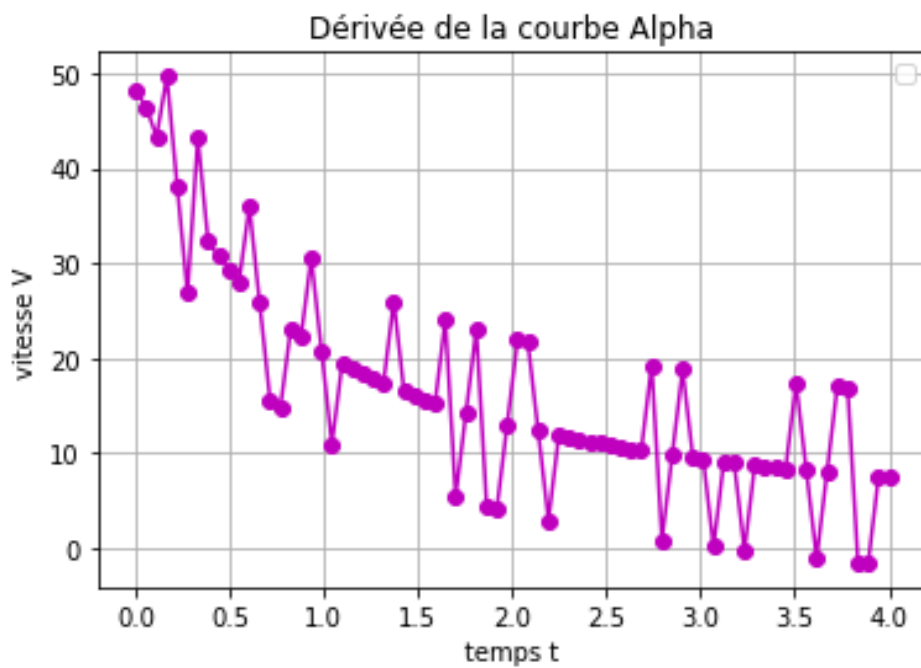


Fig 15

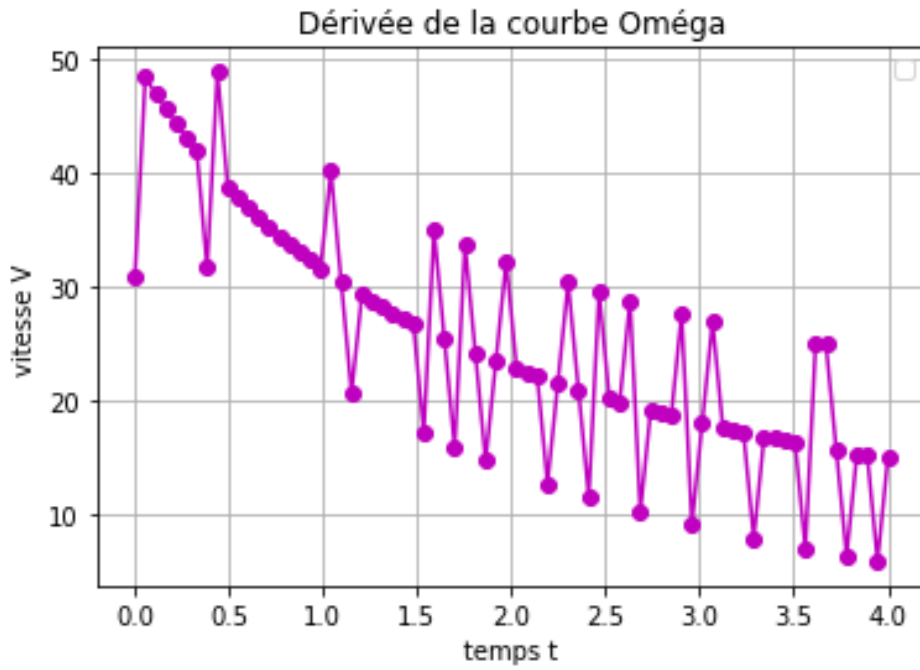


Fig 16

Et pour utiliser un modèle de régression linéaire on trace l'inverse de la vitesse (qui serait par la relation (3) affine) :

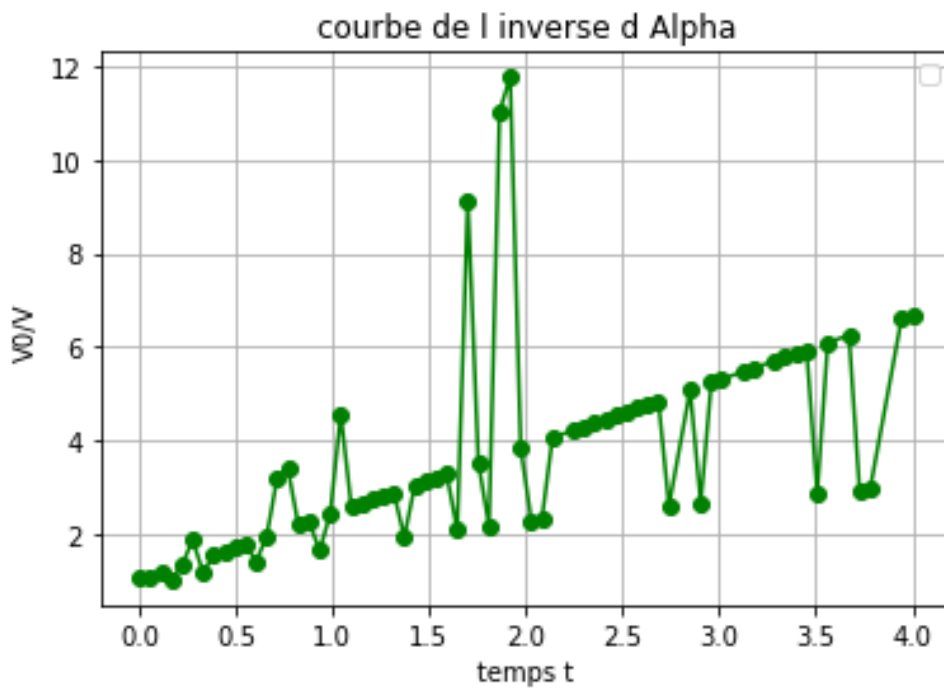


Fig 17

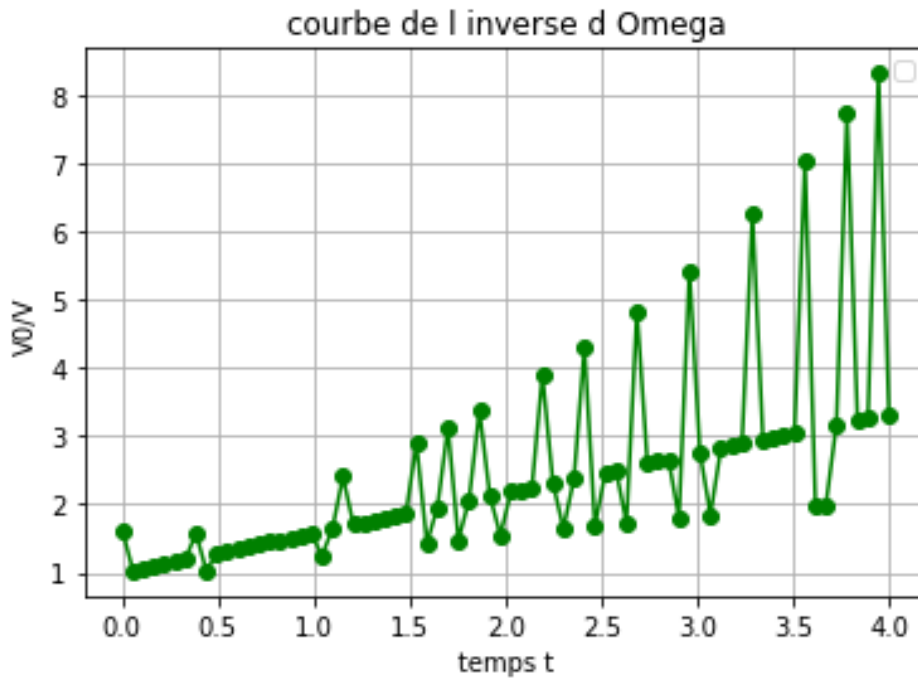


Fig 18

On utilise la régression linéaire sur python pour retrouver les valeurs des coefficients de frottement :

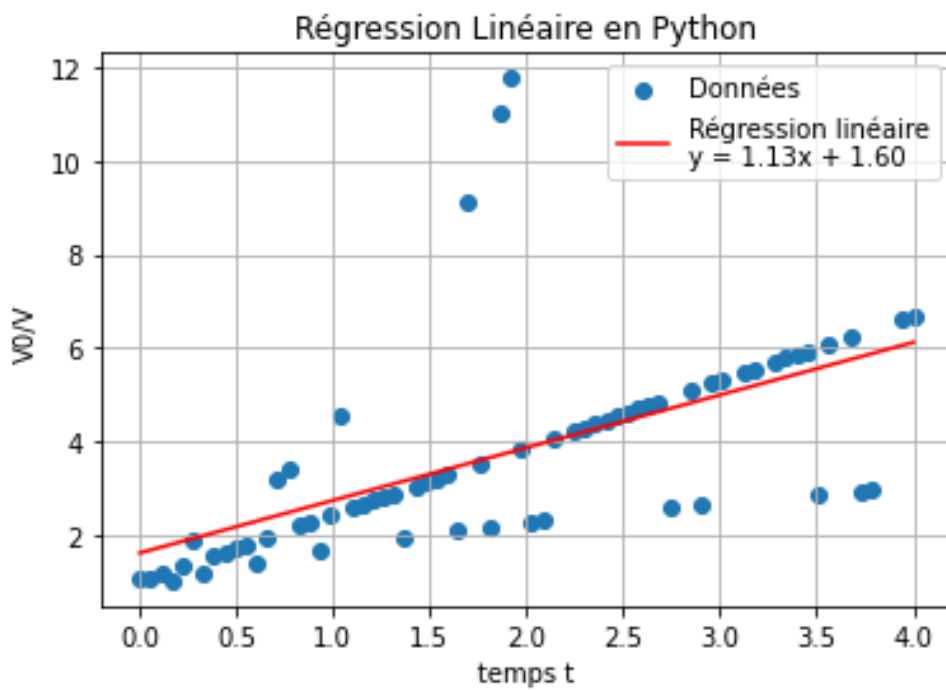


Fig 19

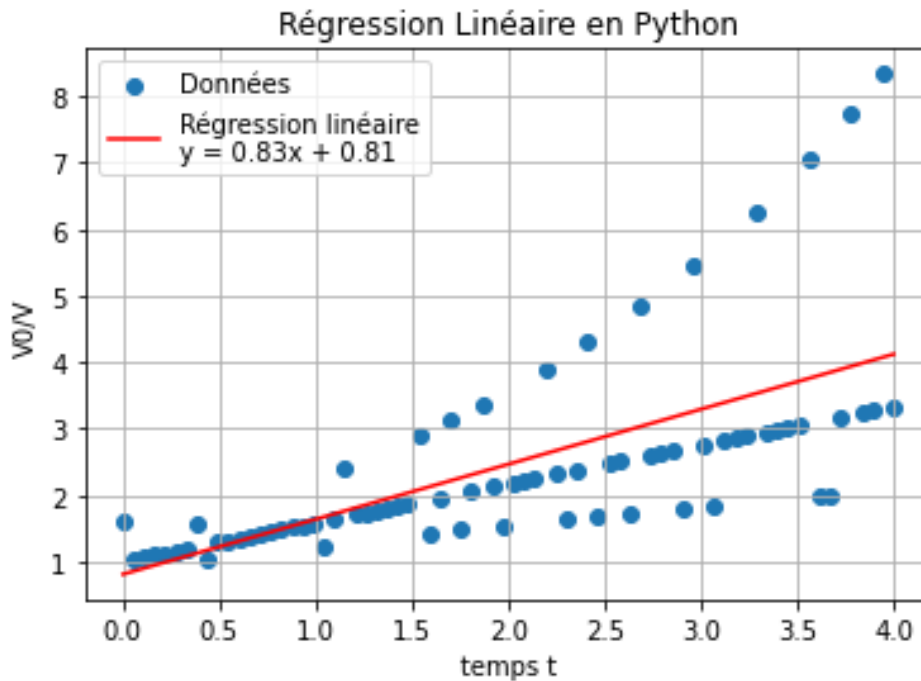


Fig 20

En prenant comme valeurs :

$$\rho = 10^3 \text{ kg/m}^3$$

$$M_1 = 2 \text{ kg}$$

$$M_2 = 1,3 \text{ kg}$$

$$V_0 = 0,5 \text{ m/s (estimé à l'aide des courbes de vitesse obtenus)}$$

On retrouve (en  $\text{m}^2$ ) :

- Alpha:  $7,1 \cdot 10^{-3}$
- Omega:  $5,97 \cdot 10^{-3}$

Ce qui est un bon ordre de grandeur. Le coefficient inhérent à Alpha plus grand que celui d'Oméga, ce qui est logique car Oméga plus massif et « large » qu'Alpha.

Une coque à bouchain semble en fin de compte bien adéquate pour une coque de kayak car minimise l'aire frontale et les dimensions au sens large du terme du kayak et donc diminue l'effet de la force de trainée et donc le coefficient de frottement comme vu précédemment.

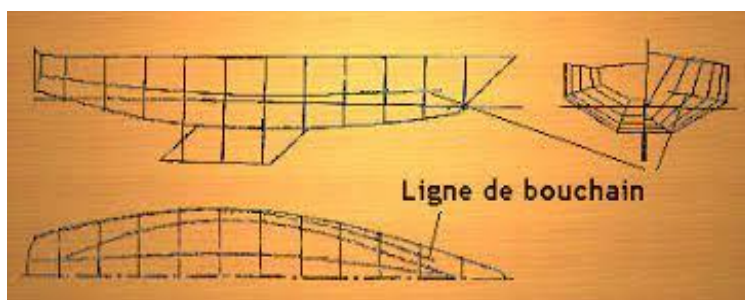


Fig 21

#### 4-Critique et amélioration du modèle :



Les courbes d'abscisse obtenus présentent des fluctuations qui s'apparentent encore plus dans les courbes de vitesse (rappelons qu'elles sont obtenues indirectement par calcul discret de dérivée).

On pourrait expliquer cela par l'incertitude sur la position du capteur ultrason qui va de 5% à 13%. De plus le modèle est simple et ne tient pas compte de l'action des vagues. De plus rien n'assure que l'élan donné est le même pour les 2 modèles et le câble relié à la carte Arduino gêne un peu et peut légèrement influencer les résultats.

## 5-Annexe :

### 5-1 Code Arduino :

```

sketch_oct31a
int trig = 3 ;
int echo = 2 ;
long duree;
float distance ;

void setup() {
  pinMode(trig, OUTPUT );
  pinMode(echo, INPUT );
  Serial.begin(9600);
}

void loop() {
  // impulsion de 10 us sur la broche trig => déclenchement de la mesure
  digitalWrite(trig, 1 );
  delayMicroseconds(10);
  digitalWrite(trig, 0 );
  // lecture de la durée d'état haut de l'impulsion en us
  duree = pulseIn(echo, HIGH);
  distance = 0.017 * duree ;
  Serial.print("distance en cm : ");
  Serial.println(distance);
  delay(10);
}

```

### 5-2 Code Python de l'abscisse : (Fig 13 et Fig 14)

```

5 @author: yahya
6 """
7
8 alpha=[(0.0, -1.0), (0.0547945205479452, 1.6377826632223518),(0.1095890410958904, 4.090624952083782),
9 omega=[(0.0, 0.0), (0.0547945205479452, 1.6967464490105058), (0.1095890410958904, 5.3110294476641755),
10
11 import matplotlib.pyplot as plt
12 import numpy as np
13 from sklearn.linear_model import LinearRegression
14
15 def tracer_courbe(points):
16     x = [point[0] for point in points]
17     y = [point[1] for point in points]
18     plt.plot(x, y, marker='o',color='r')
19     if points==alpha:
20         plt.title('Dérivée de La courbe Alpha')
21     else:
22         plt.title('Dérivée de La courbe Oméga')
23     plt.xlabel('temps t')
24     plt.ylabel('abscisse x')
25     plt.grid(True)
26     plt.show()
27 tracer_courbe(alpha)
28 tracer_courbe(omega)

```

**5-3 Code Python de la vitesse :** (Fig 15 et Fig 16)

```

30 def g(v):
31     #V0=50 cm/s
32     return 50/v
33
34 def dérivée(points):
35     x = [point[0] for point in points]
36     y = [point[1] for point in points]
37     y_prime = np.gradient(y, x)
38     plt.plot(x, y_prime, marker='o', color='m')
39     if points==alpha:
40         plt.title('Dérivée de La courbe Alpha')
41     else:
42         plt.title('Dérivée de La courbe Oméga')
43     plt.xlabel('temps t')
44     plt.ylabel('vitesse V')
45     plt.legend()
46     plt.grid(True)
47     plt.show()
48     dérivée(alpha)
49     dérivée(omega)

```

**5-4 Code Python de l'inverse :** (Fig 17 et Fig 18)

```

51 def condition(d):
52     for e in d:
53         if e<0 or cnd(e,d):
54             return False
55     return True
56 def cnd(r,points):
57     return np.abs(r)>12
58 def inverse(points):
59     x = [point[0] for point in points]
60     y = [point[1] for point in points]
61     y_prime = list(np.gradient(y, x))
62     inv=list(map(g,y_prime))
63     #on utilise un filtre de valeur
64     while not condition(inv):
65         for elt in inv:
66             if elt<0 or cnd(elt,points):
67                 x.remove(x[inv.index(elt)])
68                 inv.remove(elt)
69     plt.plot(x, inv, marker='o', color='g')
70     if points==alpha:
71         plt.title('courbe de L inverse d Alpha')
72     else:
73         plt.title('courbe de L inverse d Omega')
74     plt.xlabel('temps t')
75     plt.ylabel('V0/V')
76     plt.legend()
77     plt.grid(True)
78     plt.show()
79     inverse(alpha)
80     inverse(omega)
81

```

**5-5 Code python de la régression linéaire :** (Fig 19 et Fig 20)

```

82  def régressionlinéaire(points):
83      x = [point[0] for point in points]
84      y = [point[1] for point in points]
85      y_prime = list(np.gradient(y, x))
86      inv=list(map(g,y_prime))
87      #on utilise un filtre de valeur
88      while not condition(inv):
89          for elt in inv:
90              if elt<0 or cnd(elt,points):
91                  x.remove(x[inv.index(elt)])
92                  inv.remove(elt)
93      X=np.array(x).reshape(-1,1)
94      y=np.array(inv).reshape(-1,1)
95      model = LinearRegression()
96      model.fit(X, y)
97      slope = model.coef_[0][0]
98      intercept = model.intercept_[0]
99      y_pred = model.predict(X)
100     plt.scatter(X, y, label='Données')
101     plt.plot(X, y_pred, color='red', label=f'Régression Linéaire\ny = {slope:.2f}x + {intercept:.2f}')
102     plt.xlabel('temps t')
103     plt.ylabel('V0/V')
104     plt.title('Régression Linéaire en Python')
105     plt.legend()
106     plt.grid(True)
107     plt.show()
108     régressionlinéaire(alpha)
109     régressionlinéaire(omega)

```

## 6-Bibliographie:

- [1] P.S. JACKSON : Performance prediction for Olympic kayaks : Journal of Sports Sciences, 1995, 13, 239-245
- [2] CHARLIE PRÉTOT, RÉMI CARMIGNIANI, LOUP HASBROUCQ, ROMAIN LABBÉ, JEAN-PHILIPPE BOUCHER ET CHRISTOPHE CLANET : On the Physics of Kayaking :  
<https://www.mdpi.com/journal/applsci>
- [3] SUMNER, D., SPRIGINGS, E. J., BUGG, J. D., & HESELTINE, J. L. : Fluid forces on kayak paddle blades of different design : Sports Engineering, 6(1), 11–19
- [4] BONAIUTO, V., GATTA, G., ROMAGNOLI, C., BOATTO, P., LANOTTE, N., & ANNINO, G. : A New Measurement System for Performance Analysis in Flatwater Sprint Kayaking :  
<https://www.mdpi.com/journal/applsci>