

## Egreneur de raisin à vin : acquisition et traitement des données

### Objectifs

Après acquisition des données de vitesse et de couple sur un système vibrant, par un traitement de donnée sous Python :

- Lisser les données pour les exploiter
- Utiliser la fonction d'intégration
- Utiliser la fonction de dérivation

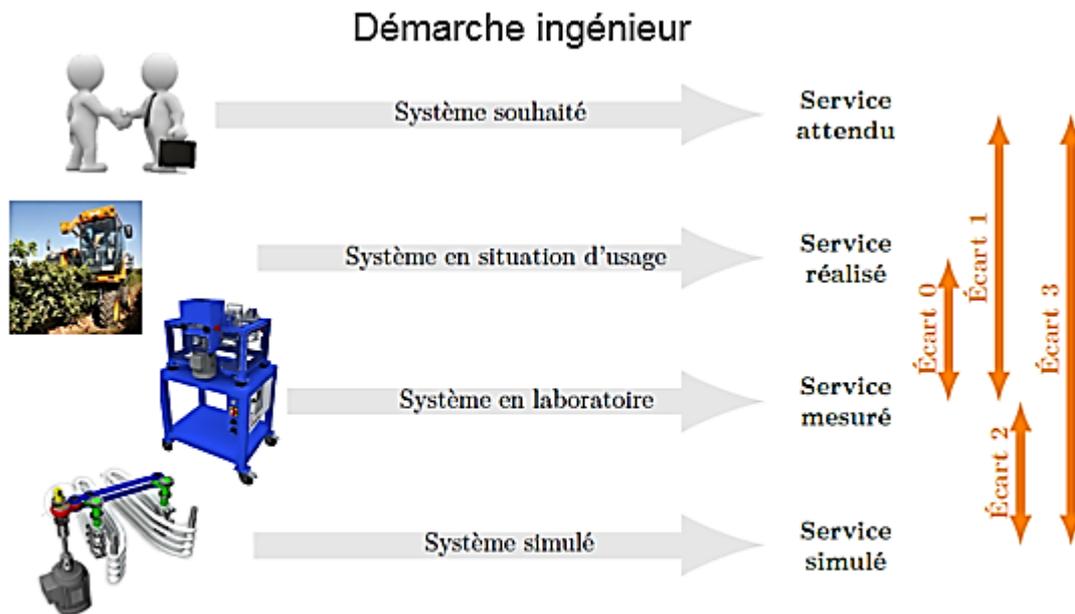


FIGURE 1 – Démarche de l'ingénieur centrée sur la mesure des écarts.

**Objectif : minimiser les écarts**

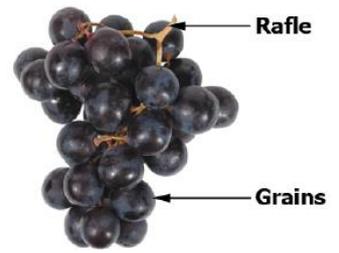
### AVERTISSEMENT

**VOUS DEVEZ DEPLACER TOUT DOCUMENT NUMERIQUE MODIFIABLE DANS UN DOSSIER PERSONNEL AVANT OUVERTURE ET MODIFICATION. IL SAGIT ICI NOTAMMENT DU FICHIER PYTHON FOURNI.**

## Contexte

Dans le processus d'élaboration du vin, l'égrenage constitue la première étape, juste après la récolte. Elle consiste à séparer les grains de la rafle et à ne conserver que les plus sains. Les grains verts ou gâtés sont éliminés pour ne pas nuire à la qualité du vin. La rafle est souvent, elle aussi, éliminée.

L'opération est réalisée par une machine installée au chai\* en début de la chaîne d'élaboration du vin (systématiquement dans le cas d'une vendange manuelle). Toutefois, dans le cas de la vendange mécanique, la machine peut-être embarquée directement sur la machine à vendanger, supprimant ainsi cette étape au niveau du chai.



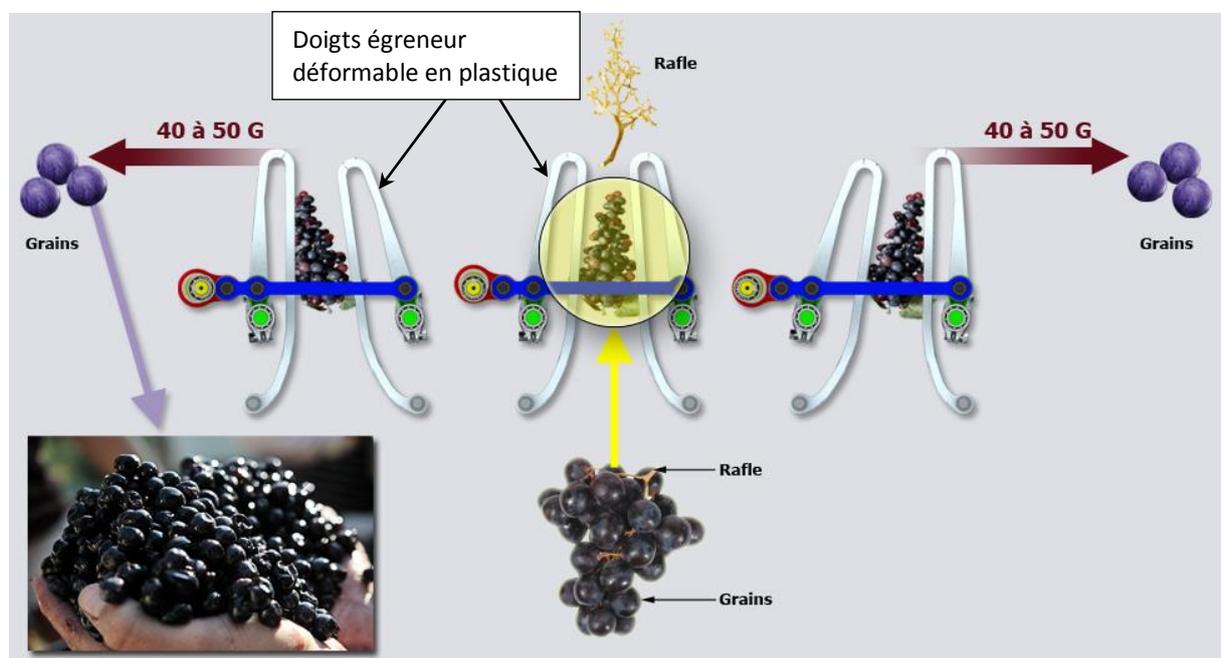
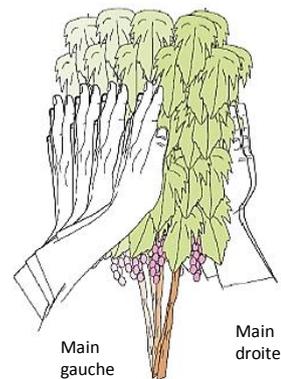
\*chai : lieu de la vinification/conservation du vin en vue de son élaboration.

- Visionner les vidéos de présentation n°1 (5min = longue : avancez éventuellement), puis n°2 (1 min).

## Fonctionnement

Les doigts égreneurs sont en plastique déformable. Ils vibrent à haute fréquence afin de séparer en douceur les grains de la rafle (inertie du fruit plus importante que la rafle). C'est un principe bien connu que l'on met parfois naturellement et manuellement en œuvre pour détacher les fruits d'une plante (voir illustration ci-contre).

Les baies ainsi égrenées passent au travers d'un tapis à claire voie (= avec des trous !) tandis que les rafles sont emmenées par le convoyeur. Les baies sont amenées sur une table de tri à rouleaux pour séparer les grains non désirés.



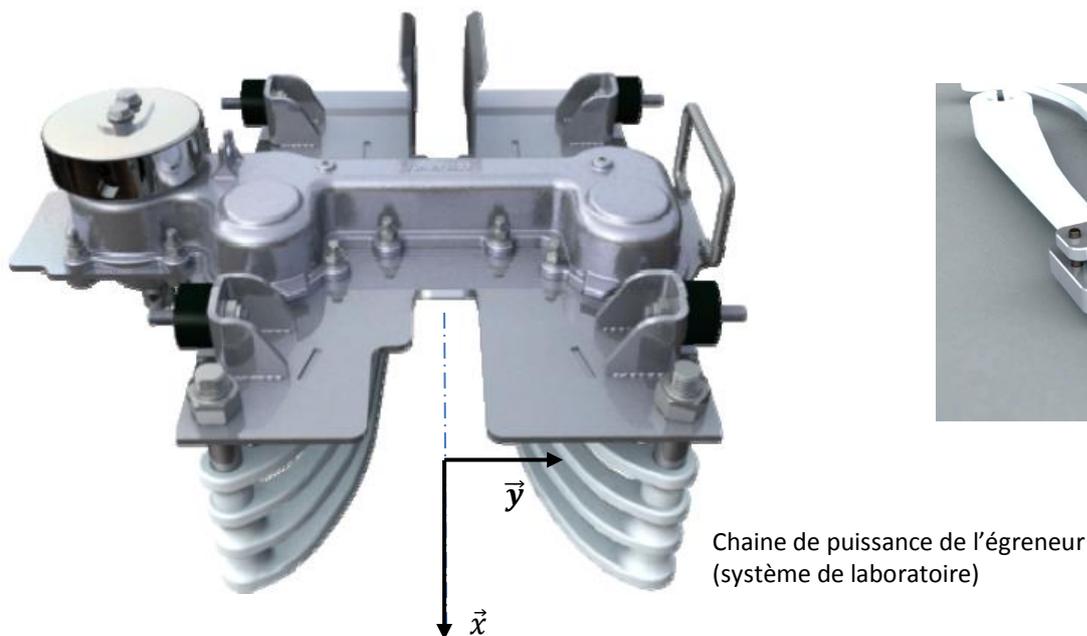
## Préparation à faire chez soi avant la séance de TP

- Lire tout le sujet
- Visionner les deux vidéos attentivement
- Faire l'IBD de la chaîne de puissance de l'égreneur montrant la transmission de mouvement du moteur jusqu'aux grappes de raisin
- Décrire le principe de la transformation de mouvement.



### Problématique de l'activité pratique proposée

Les illustrations ci-dessous montrent la chaîne de puissance telle que visible sur l'égreneur de laboratoire. Une des huit lames est équipée d'un capteur accéléromètre.

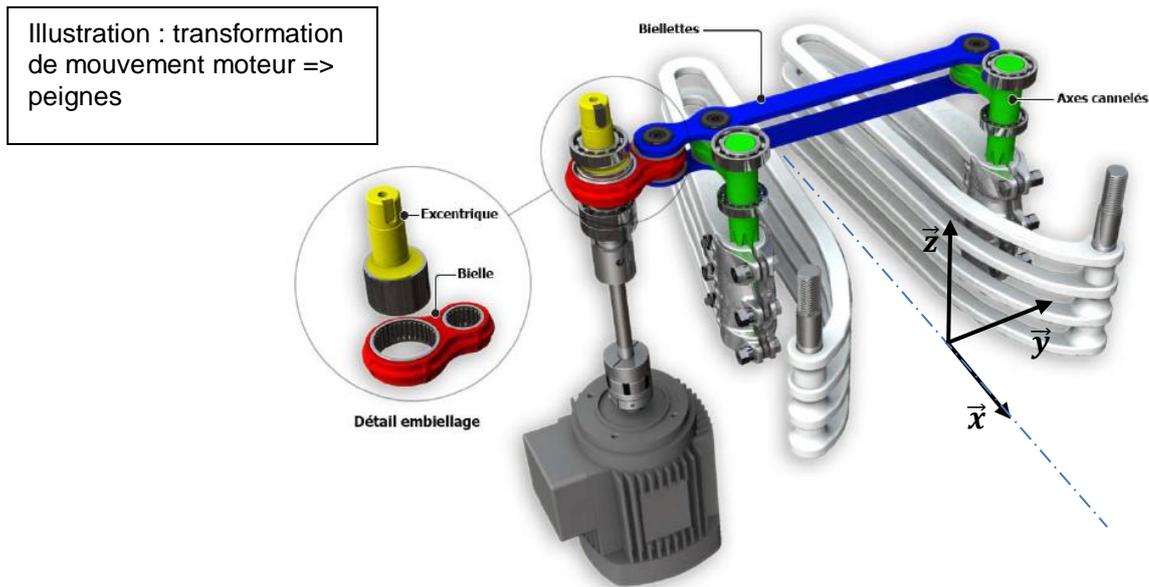


L'objectif de cette activité pratique est d'analyser et traiter les données acquises lors d'essai sur le système en vue d'une analyse des conséquences des vibrations sur le couple et la vitesse du rotor.

### 1<sup>ère</sup> manipulation : mise en route - première constatation (10 min)

- Référez-vous à l'annexe 1 de mise en route du système en fin de document et le faire fonctionner.
- Machine à l'arrêt, ouvrez le capot métallique supérieur : faites tourner à la main le volant d'inertie, regarder le mouvement des lames.
- Vibrations : selon quel axe prédominant se produisent-elles ? A la vue de l'illustration ci-après expliquer leur origine. Les axes  $\vec{x}$ ,  $\vec{y}$  et  $\vec{z}$  sont indiquées sur l'illustration.





## 2<sup>ème</sup> manipulation : pilotage par bus de donnée et acquisition (10 min)

Référez-vous à l'annexe 2 pour cette manipulation : pilotage Bus et acquisition/sauvegarde.

- Piloter l'égreneur depuis le PC via le bus de données vers le variateur ATV.
- Vitesse de rotation moteur : 1000 tr/min.
- Bien lire l'annexe 2 jusqu'au bout : acquisition et vue du tableau de données acquises.
- Si l'acquisition ne fonctionne pas (elle est un peu capricieuse sur ce système), n'insistez pas : les données sont fournies sous forme d'un tableau. Vous pouvez passer à la partie « traitement informatique » directement.
- → Allez sur un PC « informatique ». Un étudiant par poste.



*20 minutes **maximum** ont du s'écouler depuis le début de la séance. Si vous avez dépassé ce délai, vous êtes très en retard et avez besoin d'aide : appelez le professeur.*

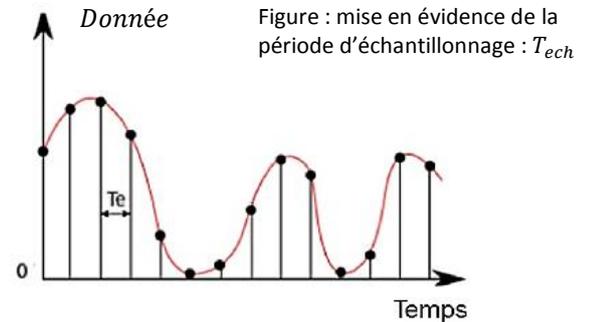


## Présentation du tableau de données avant lecture par Python

Travail individuel maintenant : un étudiant par PC.

- **Voir en annexe** l'allure du tableau de valeurs : ne pas ouvrir le fichier !

- Regarder les instants du tableau de valeurs (colonne « temps ») : quelle est la période d'échantillonnage  $T_{ech}$  de l'acquisition numérique des données de l'égreneur ? déduire la fréquence d'échantillonnage  $f_{ech}$ .

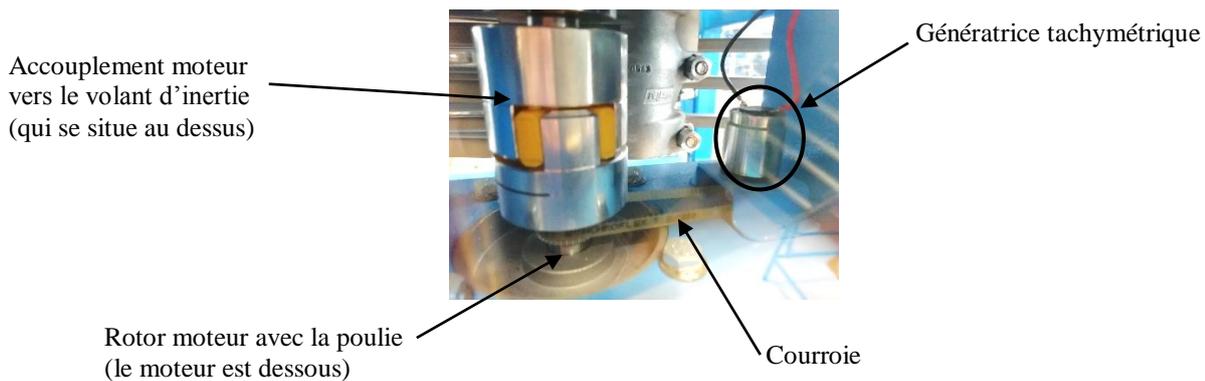


- Quel est le numéro d'ordre des colonnes « temps » et « vitesse » du point de vue Python.

## Intégration de données : angle balayé par le rotor moteur (20 min)

Le système est équipé d'une génératrice tachymétrique mesurant la vitesse du rotor moteur.

Elle est entraînée par une poulie/courroie crantée. → Allez jeter un coup d'œil...



C'est grâce à ce capteur qu'on acquière la vitesse du rotor.

D'un point de vue pratique il peut être nécessaire de connaître l'angle balayé par le rotor moteur. Par exemple, pour la maintenance du système (entretien), connaître le « nb de tours » de l'arbre moteur à un moment de sa vie permet de déclencher certaines opérations d'entretien prédictives telles que le remplacement des roulements.

Par définition, l'angle balayé entre deux instants est calculé par intégration de la vitesse instantanée  $\omega_{mot}(t)$  :  $\theta(t_1 \text{ à } t_2) = \int_{t_1}^{t_2} v(t). dt$ .

Ouvrir le code Python «Egreneur-trapeze\_vitesse».

Modifiez le code : saisir les données manquantes.

```

10 #####
11 #
12 # # EGRENEUR MESURES TRAITEMENT DES DONNEES
13 #
14 #####
15 # chemin d'accès du dossier à définir
16 os.chdir(r" .....SAISIR LE CHEMIN DU FICHIER.....")
17 # Ouvrir le fichier des points simulés
18 fichier = open("Essai_egreneur.csv", "r")
19
20 # Lire la première ligne : nom des variables et unités
21 variable1 = fichier.readline().rstrip('\n\r').split(";")
22
23
24 donnees = ligne.rstrip('\n\r').split(";")
25 # Créer de la liste des points
26 Temps.append(float(donnees[...])) #n° colonne du temps
27 Coupl.append(float(donnees[...])) #n° colonne du couple
28 Vit.append(float(donnees[...])) #n° colonne de la vitesse moteur
29 Accel.append(float(donnees[...])) #n° colonne de l'accélération lame
30 fichier.close()

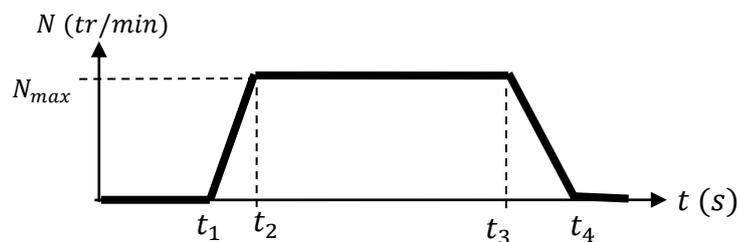
```

- Compléter la ligne ci-dessous pour visualiser la vitesse moteur

```
plt.plot(., ., '-', label='Vit') #A SAISIR#####
```

- Lancer le code. La vitesse moteur s'affiche.

Il est visible que l'évolution de la vitesse est une évolution classique « en trapèze » en trois phases : accélération/vitesse constante/décélération.



- Relever les données  $t_1, t_2, t_3, t_4, N_{max}$ .

Le script Python dédié à l'intégration par la méthode de calcul des petits trapèzes est déjà dans le code. Ce sont les lignes reproduites ci-dessous.

```
66 ##calcul aire avec integration (methode petits trapezes)
67 #integration vitesse pour avoir l'angle balayé en rad
68
69 #Fonction générale de calcul d'aire
70 def calcul_aire(y,t): #creation de la fonction integration de la liste de sortie y sur la liste des temps t
71     aire=[0]
72     for i in range(len(t)-1):
73         aire.append((aire[i]+(y[i]+y[i+1])*0.5*(t[i+1]-t[i])))
74     return aire
75
76 Angle=calcul_aire(Vit,Temps) #calcul de l'angle grâce à la fonction intégration calcul_aire(y,t)
77
78 Angletr=[0]
79 for k in range(len(Angle)-1):
80     Angletr.append(Angle[k]/60)
81
```

- Quel est le rôle des lignes 78, 79, 80 ?
- Ajouter la ligne de code (*plt.plot etc...*) qui permet d'afficher l'angle, en plus de la vitesse déjà affichée.
- Lancer le code.

Vous constatez que La courbe d'évolution de l'angle comporte aussi trois parties, correspondant à l'évolution de la vitesse.

- Expliquer en quoi le graphe de l'évolution de l'angle correspond bien à l'évolution de la vitesse sur chacun des trois intervalles de temps :  $[t_1, t_2]$ ,  $[t_2, t_3]$ ,  $[t_3, t_4]$  . Pensez « dérivée ».
- Quel est l'angle total balayé par le rotor en fin d'essai  $\theta_{total}$  donné par la courbe des angles ?
- Calculer l'angle balayé par le rotor par « calcul d'aire de surface » basique sur le modèle trapèze donné précédemment et pour lequel vous avez les instants  $t_i$  (comme on a fait en TD de cinématique).
- Comparer les deux valeurs d'angle : celles obtenues avec le code Python et celles du « pur » modèle trapèze avec aire des surfaces.

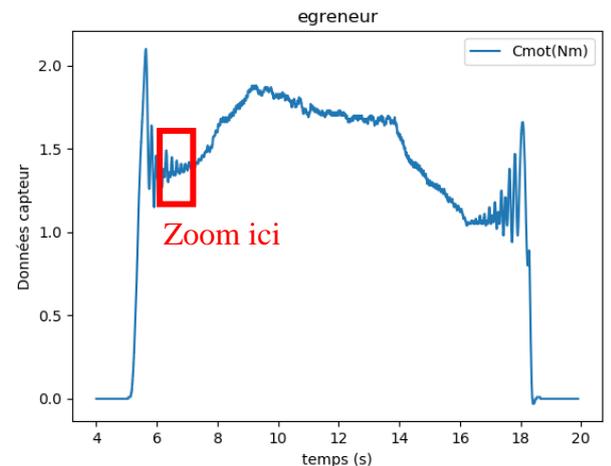
## Affichage/Lissage des données : évolution du couple rotor moteur (30 min)

Modifiez le code pour :

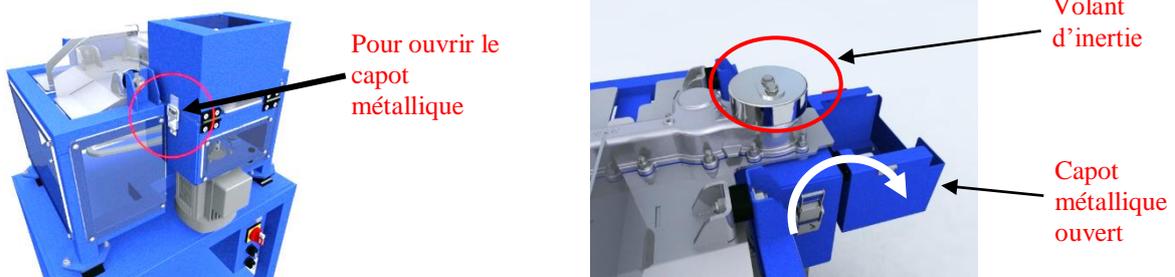
- ne plus afficher les courbes de vitesse et angle
- afficher la courbe de couple (ordonnée) en fonction du temps (abscisse)

Lancer le code.

- Agrandir l'écran et zoomer sur la zone d'oscillation du couple moteur indiquée.
- Quelle est la période de ces oscillations  $T_{Cmot}$  ?  
Déduire la fréquence  $f_{Cmot}$ .



→ Voyagez : pour connaître la raison de ces fortes oscillations de couple, allez sur le système, ouvrez le capot métallique supérieur et tournez le volant d'inertie cylindrique à la main. Regardez les lames.



Constatez les variations de couple ressenties à la main.

- En plus des variations de couple dues aux effets d'inertie dus au mouvement alternatif des pièces et identifiés précédemment, d'où proviennent les variations de couple que vous ressentez à la main ?

On souhaite lisser la courbe de couple, surtout dans cette zone zoomée intéressante. Deux méthodes numériques sont couramment utilisées : par moyenne glissante et avec un filtre passe bas.

On va s'intéresser à un filtrage passe bas.

### Filtrage passe bas : principe

On applique à l'entrée à filtrer la fonction de transfert du premier ordre :

$$S(p) = \frac{1}{1+\tau p} E(p)$$

avec la pulsation de coupure à -3dB du filtre qui est :  $\omega_c = \frac{1}{\tau}$ .

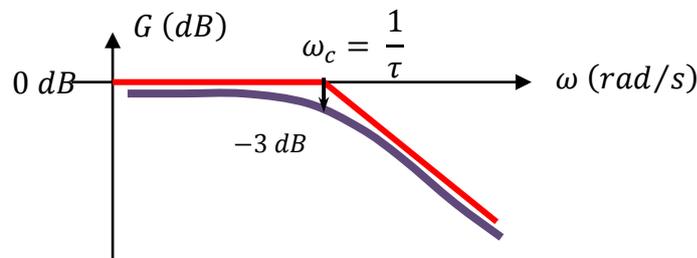


Figure : diagramme de Bode gain d'un filtre passe bas

La fréquence de coupure est donc :  $f_c = \frac{1}{2\pi\tau}$ .

Si on souhaite une fréquence de coupure  $f_c$ , la constante de temps du filtre nécessaire est donc :  $\tau = \frac{1}{2\pi f_c}$ .

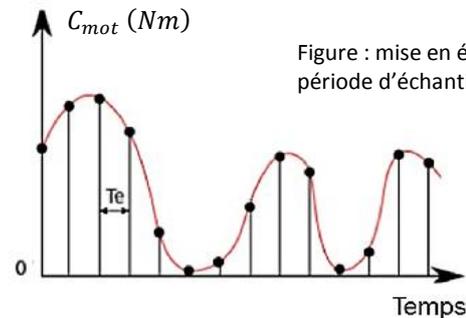
Dans le domaine temporel, et en tenant compte du fait que le signal est numérisé avec une période d'échantillonnage  $T_{ech} = t_{n+1} - t_n$ , où  $t_n$  est l'instant à l'indice de prise de mesure  $n$ , il vient :

$$\Leftrightarrow \tau \cdot \frac{ds(t)}{dt} = e(t) - s(t)$$

$$\Leftrightarrow \tau \cdot \frac{s_{n+1} - s_n}{t_{n+1} - t_n} = e_n - s_n$$

$$\Leftrightarrow \tau \cdot \frac{s_{n+1} - s_n}{T_{ech}} = e_n - s_n$$

$$\Leftrightarrow \boxed{s_{n+1} = s_n + \frac{T_{ech}}{\tau} (e_n - s_n)}$$

Figure : mise en évidence de la période d'échantillonnage :  $T_{ech}$ 

Cette expression donne la sortie calculée à l'instant  $n+1$  en fonction de la sortie et l'entrée « précédentes » à l'instant  $n$ .

Cette dernière expression est utilisée dans un code Python pour filtrer le signal.

Le code est le suivant :

```
##### Code python filtrage avec filtre passe bas #####
```

```
e=liste #choix de la donnée à lisser (liste de valeur extraite de l'acquisition)
```

```
T=0.1 #constante de temps du filtre
```

```
h=0.0 #pas de calcul (= periode d echantillonnage)
```

```
s=[e[0]]
```

```
for n in range(len(e)-1):
```

```
    s.append(s[n]+(h/T)*(-s[n]+e[n]))
```

```
# frequence de coupure du filtre, juste pour information
```

```
fc=1/(2*3.1416*T) ## T est la constante de temps du filtre
```

```
print('freq de coupure du filtre (Hz) : ',fc)
```

```
##### fin du code #####
```

Insérer le code de lissage dans le code précédent (copier/coller). Faire les modifications nécessaires.

Choix de la constante de temps du filtre.

- Choisir  $\tau = \tau_{cmot}$  correspondant à la fréquence des oscillations de couple à filtrer  $f_{cmot}$  identifiée précédemment : donner la valeur de  $\tau = \tau_{cmot}$ .
- Quelle valeur exacte saisissez-vous pour la période d'échantillonnage  $T_{ech}$  ?

Afficher deux courbes : la courbe de couple brute ET la courbe de couple lissée. Augmenter la taille de fenêtre en plein écran. Zoomer la zone intéressante : les oscillations sont atténuées (normal pour du lissage !).

- Vue la fréquence de coupure du filtre choisie (qui est la fréquence des oscillations donc) : de quelle gain (dB) l'amplitude des oscillations est-elle atténuée ? (pas de calcul ici, juste de la réflexion)

Le lissage n'est pas suffisant. On veut encore atténuer les oscillations.

- Pour cela, faut-il augmenter ou diminuer la fréquence de coupure du filtre  $f_c$  ?

Lancer un nouveau filtrage : prendre une valeur de  $\tau = 4 \times \tau_{cmot}$ .

- Visualiser la courbe de filtrage, zoomer. Conclusion.

Augmenter encore la valeur de  $\tau$

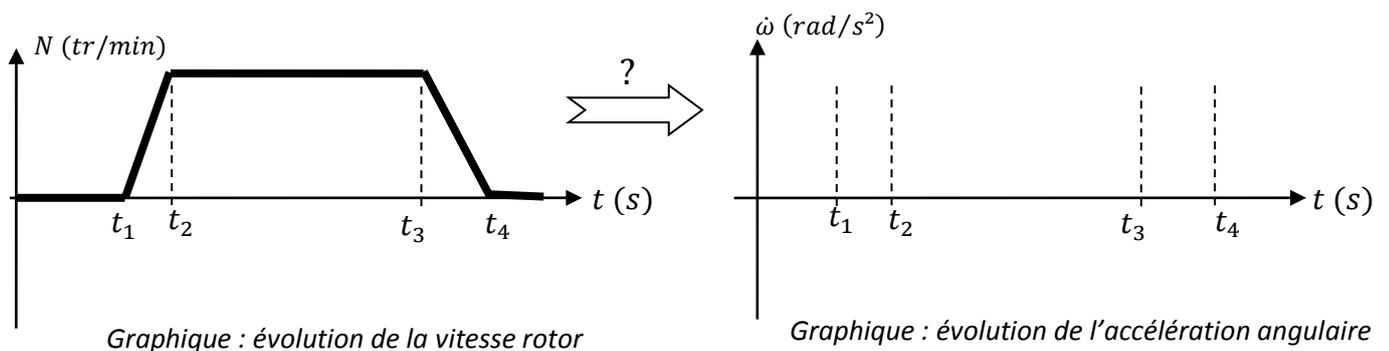
- Visualiser la courbe locale (zoom) et globale : conclusion ?  
Augmentation de la constante de temps du filtre passe bas (donc diminution de la fréquence de coupure du filtre) : avantage/inconvénient ?

### Dérivation de données : accélération du rotor (20 min)

On peut avoir besoin de connaître l'accélération angulaire  $\dot{\omega}(t)$  du rotor pour traiter des problèmes dynamiques.

On va donc dériver la vitesse du rotor.

- Vue la loi en trapèze : quelle devrait être la loi d'accélération ?



Le code Python pour dériver temporellement est le suivant :

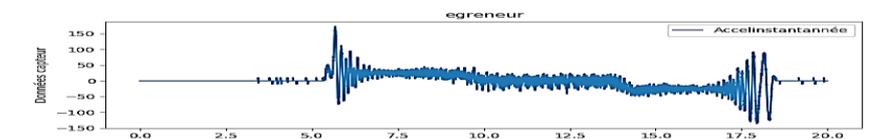
```
##### Code Python dérivation par taux d'accroissement local #####
# dérivation vitesse pour accél instantanée rotation en rad/s²
def der(y,t,indice): #creation de la fonction derivation de la liste de sortie y par rapport à t à
l'indice k
    return (y[indice]-y[indice-1])/(t[indice]-t[indice-1])
derVit=[0]
for k in range (1,len(Temps)): #creation de la liste de dérivée de la donnée egreur Vitesse
angulaire/temps
    derVit.append(3.1416*der(Vit,Temp,k)/30)

##### Fin du code #####
```

Copier/coller ce code dans le script Python

Faire les modifications du code.

Afficher l'accélération : vous constatez que le signal est extrêmement bruité. Il est donc inexploitable. En outre nous sommes bien loin de la loi d'accélération évidente à déjà visualisée.



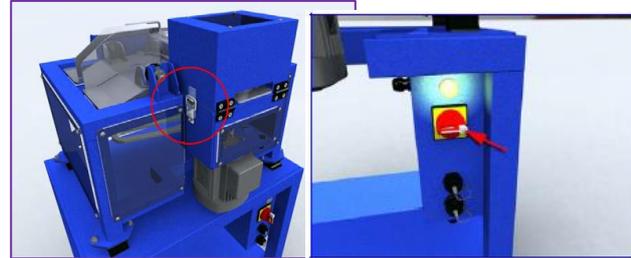
Pour remédier à cela : lisser le signal de vitesse initial et dérivez ensuite. Modifier le code. Trouver la constante de temps du filtre qui permet un lissage optimal : ni trop fort, ni insuffisant. Vous devez voir l'allure d'accélération attendue.

## FIN DE L'ACTIVITÉ PRATIQUE

## ANNEXE 1 : MISE EN ROUTE DU SYSTEME - MARCHE NORMALE

Il s'agit de la mise en route comme sur le système réel, c'est-à-dire en actionnant directement le variateur.

- Vérifier que le capot latéral transparent est fermé.
- Vérifier que le capot métallique supérieur est fermé.
- Mettre le système sous tension : gros bouton tournant quart de tour.



- Tourner le bouton de mise en marche situé sur le variateur : c'est parti.

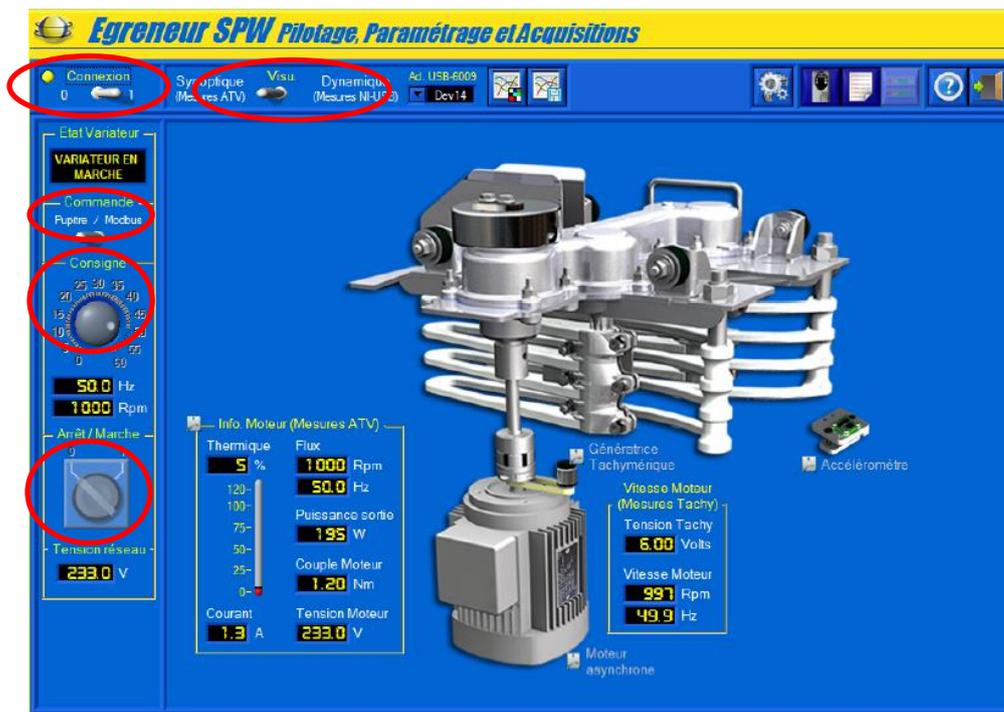


## ANNEXE 2 : PILOTAGE PAR LE BUS DE DONNEES ET ACQUISITION

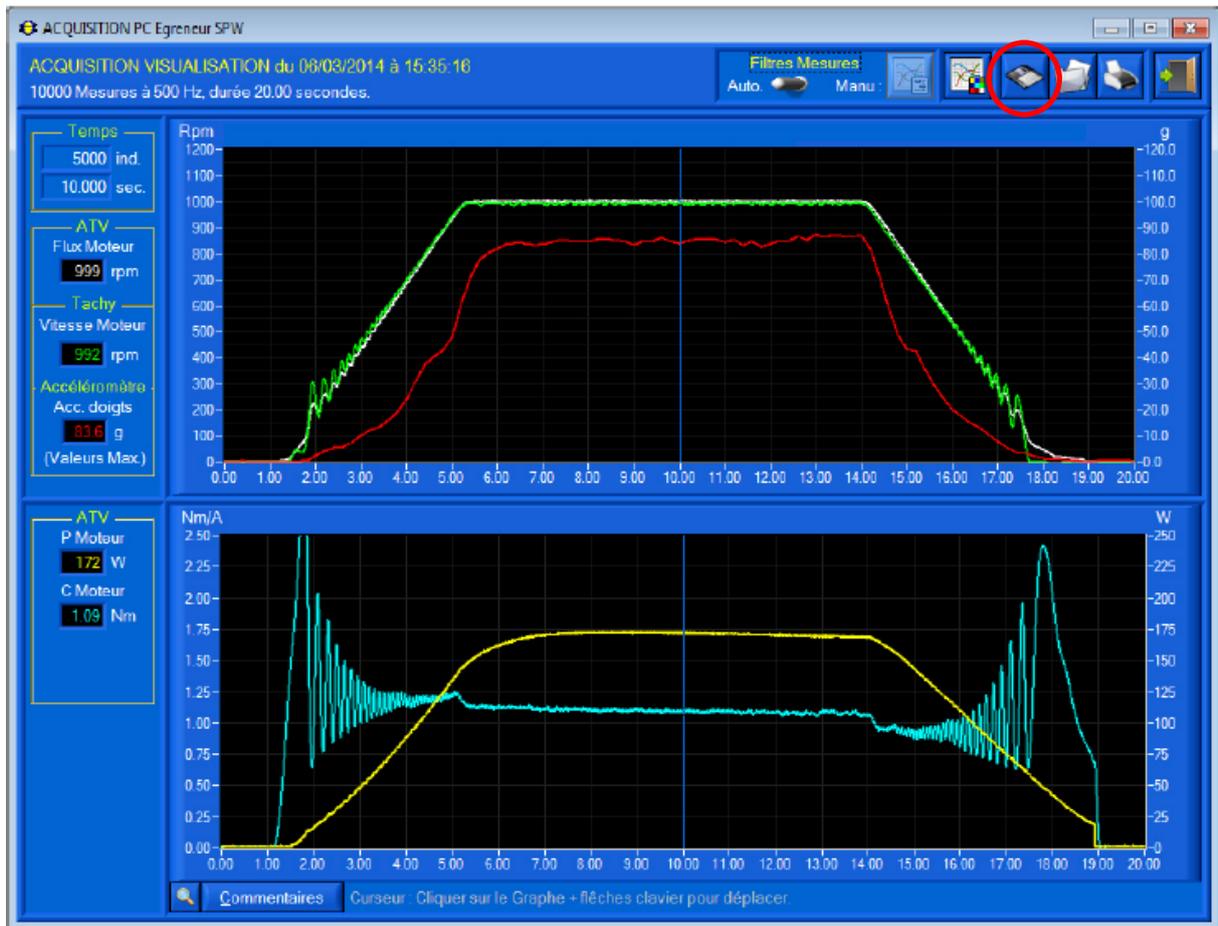
Il s'agit de piloter le système par l'appli PC : 

Cela offre des possibilités bien plus étendues que la mise en route par le variateur, comme par exemple régler la vitesse de l'arbre moteur et gérer l'acquisition des données.

- Ouvrir l'appli de l'égreneur SPW 
- Bouton « connexion ATV 0/1 » : connectez l'appli au variateur, position 1 
- Bouton sélection pupitre/Modbus : choisir modbus
- Consigne (bouton tournant) : régler 1000 tr/min
- Bouton tournant 0/1 : mettre en position 1 (marche), ça tooooooourne !



- Pour afficher les courbes : bouton en haut « Visu », choisir « Dynamique ».
- Lancer le système : bouton tournant Arrêt/Marche sur la position (1).
- Pour figer la courbe qui déroule en temps réel et stopper l'acquisition, cliquer « Acquisition mesures courantes » : la courbe est figée. Vous pouvez visualiser les données capteurs à un instant en déplaçant le curseur temporel vertical. 



- Il est possible de sauvegarder toutes les données capteurs acquises pendant l'édition de la courbe. Cette sauvegarde se fait sous la forme d'un fichier « .csv ».

Pour vous faire gagner du temps, la sauvegarde est déjà faite : le fichier est « *Essai\_egreneur.csv* ». Il est déjà disponible.

Une copie d'écran partielle du fichier est donnée ci-après, regardez-là :

- 7 colonnes en tout
- 5 colonnes de données (flux stator moteur en *Hz*, Puissance en *W*, couple moteur en *Nm*, vitesse moteur en *tr/min*, accélération du bout de la lame en *g* )
- 1 colonne pour le temps
- 1 colonne index = n° de la mesure

Note : dans Python les numéros de colonne de donnée vont de 0 à 6 (pas de 1 à 7). C'est une manie chez Python de commencer à 0 et pas à 1 !

ACQUISITION VISUALISATION du 31/01/2023 à 17:24:24						
10000 Mesures à 500 Hz, durée 20.00 secondes.						
COMMENTAIRE :						
N° MESURE		ATV :			Tachy :	Accéléromètre :
Index n°	Temps (s)	Flux (Hz)	P Mot. (W)	C Mot. (Nm)	Vitesse (rpm)	Acc. (g)
4440	8,88	906,5	176	1,8	912	45
4441	8,882	907,7	175	1,8	912	45
4442	8,884	908,7	176	1,8	913	45
4443	8,886	909,6	175	1,8	914	45
4444	8,888	909,9	176	1,8	914	45
4445	8,89	909,6	176	1,8	915	45
4446	8,892	911,1	176	1,8	915	45
4447	8,894	910,5	176	1,8	916	46
4448	8,896	910,2	176	1,8	916	46
4449	8,898	910,2	177	1,79	917	46
4450	8,9	909,9	177	1,79	917	46
4451	8,902	911,4	177	1,79	918	46
4452	8,904	910,8	177	1,79	918	46
4453	8,906	912,3	177	1,79	919	46
4454	8,908	913,2	178	1,79	920	46
4455	8,91	913,9	177	1,8	921	46
4456	8,912	915,4	178	1,8	921	46
4457	8,914	915,1	178	1,8	922	46
4458	8,916	916,6	178	1,8	922	46
4459	8,918	916,6	178	1,8	922	46
4460	8,92	916,3	178	1,8	922	46