



Portail Domoticc



Objectif : étudier la géométrie du portail selon trois approches :

- théorique;
- expérimentale;
- numérique.

Comparer les résultats et discuter les sources des écarts éventuels.

1 | Présentation

L'ouvre-portail DOMOTICC est un système industriel instrumenté d'ouverture automatisée avec pilotage à distance d'un portail à deux vantaux. Il met en œuvre un système automatique d'ouverture de portail à usage grand public.

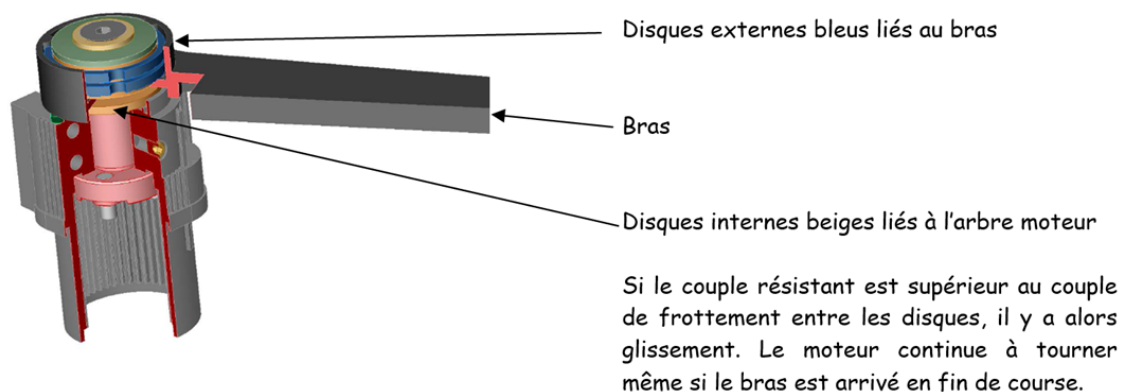
Le matériel est composé principalement d'un châssis métallique supportant le portail, d'une centrale de commande avec émetteur haute fréquence, de deux motoréducteurs avec limiteur de couple et tringlerie d'ouverture de portail et d'un ensemble de capteurs spécifiques associés à une carte de traitement.

Outre son bâti, ce système automatisé est constitué de trois sous-ensembles principaux.

1.1 Un grand vantail instrumenté

Il comprend :

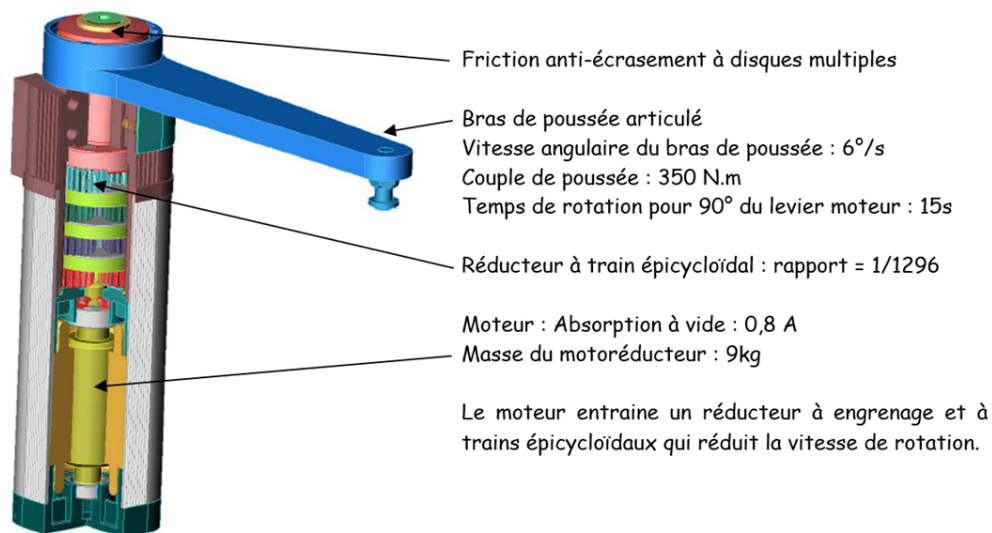
- **4 capteurs**
 - Un **potentiomètre** de précision qui permet de mesurer la position angulaire du bras du motoréducteur ;
 - Un **potentiomètre** de précision qui permet de mesurer la position angulaire du grand vantail ;
 - Une **jauge de contrainte** qui permet de mesurer le couple fourni par le motoréducteur ;
 - Une **jauge de contrainte** qui permet de mesurer le couple sur le vantail.
- Un frein permettant de simuler l'effet du vent sur le vantail ;
- Un support permettant de placer des masses de 10 kg (charge maxi 50 kg) pour modifier l'inertie du vantail ;
- Une motorisation par un motoréducteur à bras avec limiteur de couple à friction (B. F. T. - Réf. E5).



1.2 Un petit vantail

Il comprend :

- Une électro-serrure de verrouillage du portail ;
- Une motorisation par un motoréducteur à bras avec limiteur de couple à friction (B. F. T. - Réf. E5).



1.3 Un coffret de commande

Il comprend :

- Un module de commande d'ouverture de portail ;
- Les circuits de protection électrique ;
- Une alimentation 2x12 V continu ;
- Une carte de traitement des signaux capteurs.

La Figure 1 présente le diagramme de définition des blocs du portail qui vient compléter/illustrer la présentation précédente.

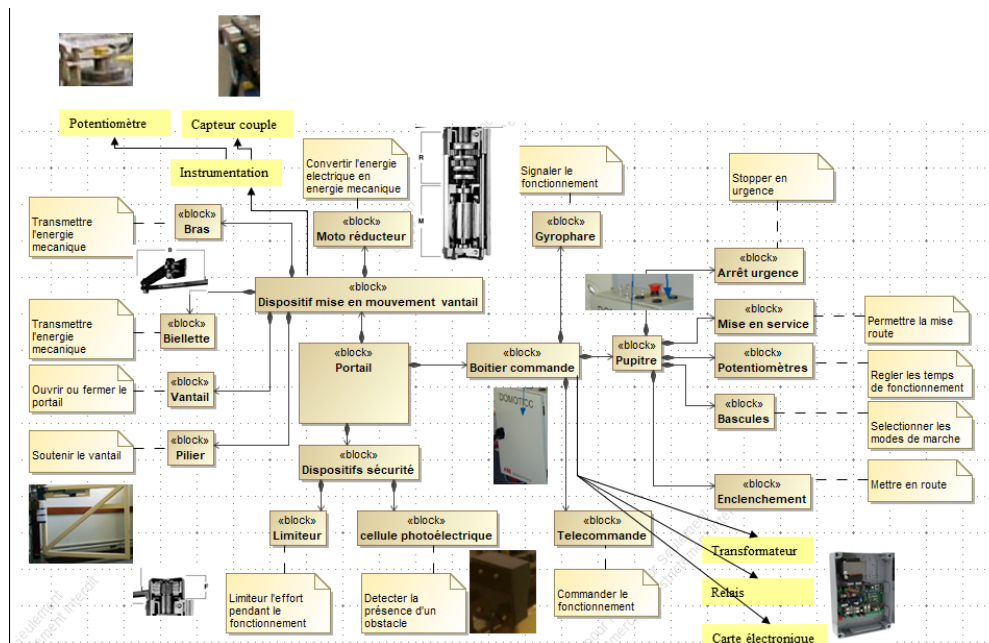


FIGURE 1 – Diagramme de définition des blocs du portail

Question 1 : A partir de la description, du diagramme de définition des blocs, et si nécessaire du système réel, compléter les chaînes fonctionnelles du portail de la Figure 2. *Ne pas écrire sur le sujet! Reproduire ou écrire numériquement.*

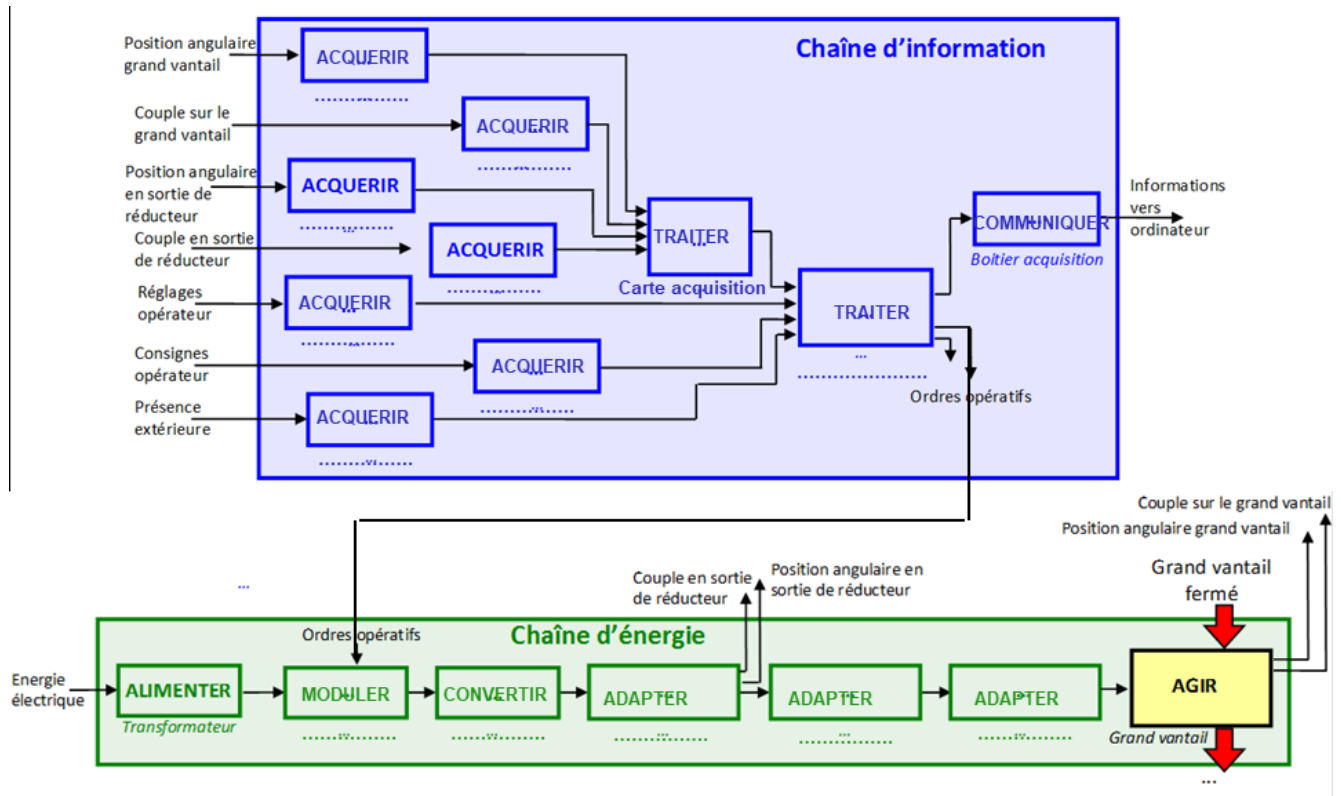


FIGURE 2 – Chaînes fonctionnelles à remplir

2 | Prise en main du système

Ne pas intervenir sur les boutons de réglage.

2.1 Procédure 1 : Fonctionnement en mode manuel sans cellule

- ⚙ Positionner l'interrupteur TCA sur HORS service (ne pas forcer, mais soulever légèrement avant de déplacer le bouton);
- ⚙ Positionner l'interrupteur IBL sur HORS service;
- ⚙ Positionner l'interrupteur FCH sur EN service;
- ⚙ Vérifier que le bouton d'arrêt d'urgence est désenclenché, mettre sous tension à l'aide du bouton M/A situé sur le coffret puis le bouton EN SERVICE;
- ⚙ Tout en maintenant enfoncé le bouton ENCLenchement,
- ⚙ Donner une impulsion sur la touche DÉMARRAGE;
Le cycle d'ouverture doit commencer.
- ⚙ Avant que le portail n'arrive en ouverture maximale, donner une 2^{ème} impulsion au bouton DÉMARRAGE;
Le portail doit s'arrêter.
- ⚙ Appuyer de nouveau sur le bouton DÉMARRAGE.
Le portail se referme.

2.2 Procédure 2 : Fonctionnement en mode manuel avec cellule

- ⚙ Maintenir le bouton ENCLenchement pendant toute la durée du cycle;
- ⚙ Donner une impulsion sur la touche DÉMARRAGE;
Le cycle d'ouverture doit commencer.
- ⚙ Avant que le portail n'arrive en ouverture maximale, mettre un obstacle devant la cellule;
Le portail doit s'arrêter.
- ⚙ Reprendre la même procédure sur la fermeture.

3 | Étude théorique

On s'intéresse dans cette étude au système de transformation de mouvement qui permet l'ouverture du grand vantail.

3.1 Modélisation

On donne sur la Figure 3 un schéma cinématique plan du système d'entraînement du grand vantail.

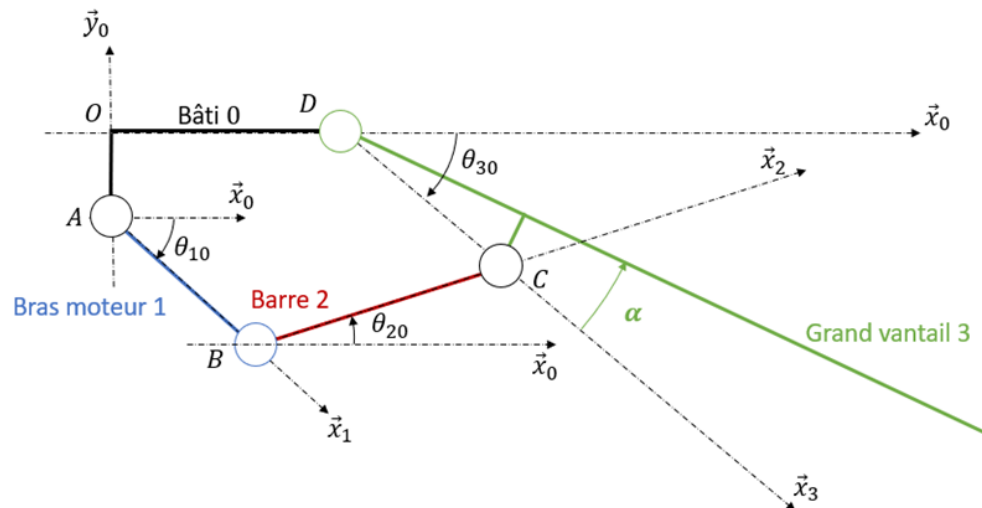


FIGURE 3 – Schéma cinématique du portail

On pose $\overrightarrow{OA} = -a\vec{y}_0$; $\overrightarrow{AB} = b\vec{x}_1$; $\overrightarrow{BC} = c\vec{x}_2$; $\overrightarrow{CD} = -d\vec{x}_3$; $\overrightarrow{OD} = e\vec{x}_0$

Données numériques (mm) : $a = 250$; $b = 280$; $c = 280$; $d = 350$; $e = 100$

Les angles sont paramétrés sur le schéma cinématique.

De par la construction, l'angle θ_{30} n'est pas l'angle du vantail par rapport au bâti. A un moment, il faudra tenir compte de l'angle α constant lié uniquement à la géométrie du vantail dans les résultats.

On donne :

$$\alpha \simeq 0,0775\text{rad} \simeq 4,44^\circ$$

3.2 Étude géométrique

Le mécanisme de transformation de mouvement étudié fait partie des systèmes 4 barres.

Ce système 4 barres est constitué de 3 barres, définies par les pièces 1, 2, 3, liées par des liaisons pivot au bâti formant la quatrième barre.

L'entrée du mouvement s'effectue au niveau de la liaison pivot entre le bâti et le bras moteur, et la sortie au niveau de la liaison pivot entre le bâti et le grand vantail.

Question 2 : Établir le graphe de liaison.

Question 3 : Réaliser les figures de changement de base

Question 4 : Par une fermeture géométrique, établir le système d'équations liant les angles θ_{10} , θ_{20} et θ_{30} aux paramètres géométriques a, b, c, d et e . En notant f_1 et f_2 des fonctions, mettre le système précédent sous la forme :

$$\begin{cases} f_1(\theta_{10}, \theta_{20}, \theta_{30}) = 0 \\ f_2(\theta_{10}, \theta_{20}, \theta_{30}) = 0 \end{cases}$$

La résolution analytique de ce système est un peu fastidieuse. Elle est présente en fin de sujet. Dans

la suite, on utilise `fsolve`, fonction du module `scipy.optimize`, pour résoudre numériquement les deux équations.

3.3 Résolution avec Python

 Ouvrir le fichier **Portail Domotic – Etudiant.py**.

Dans le système d'équations, l'angle du grand vantail est une donnée. Les inconnues sont θ_{10} et θ_{20} , regroupées dans $X = (\theta_{10}, \theta_{20})$. L'argument `arg` de `Sys` est l'angle θ_{30} utilisé dans la fermeture géométrique; il vaut `angle - alpha` pour une valeur `angle` de la liste `Theta30`.

Aide pour utiliser `fsolve`. La fonction `Sys(X, arg)` doit renvoyer les deux membres de gauche $[f_1, f_2]$: `fsolve` cherche les deux valeurs de X pour lesquelles ils sont nuls. L'import `from scipy.optimize import fsolve` et l'estimation initiale `init` sont déjà présents dans le script. Pour une valeur fixée de l'angle du modèle `t3`, l'appel s'écrit :

```
1 | sol = fsolve(Sys, x0=init, args=(t3,))
```

`sol[0]` contient θ_{10} et `sol[1]` contient θ_{20} , en radians. La virgule dans `(t3,)` est nécessaire pour former un tuple à un seul argument. Dans la boucle sur `Theta30`, prendre `t3 = angle - alpha`; pour suivre la même configuration du portail, réutiliser la solution obtenue comme estimation initiale au pas suivant.

Question 5 : Compléter le script donné de la manière suivante (les morceaux de code sont présentés un peu plus bas) :

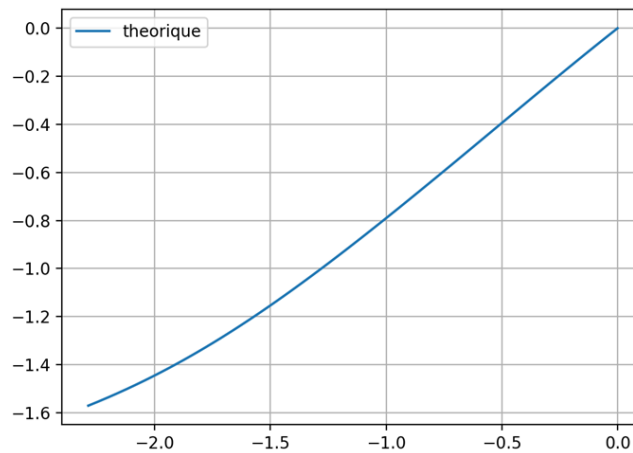
-  Définir, en utilisant par exemple la fonction `linspace`, la liste des valeurs de θ_{30} allant de -90° à 0° avec 91 points de mesure.
-  Définir la fonction `Sys(X, arg)` qui pour un θ_{30} donné, renvoie les expressions de $f_1(\theta_{10}, \theta_{20}, \theta_{30})$ et $f_2(\theta_{10}, \theta_{20}, \theta_{30})$.
-  Plus loin, à l'aide d'une boucle `for`, calculer via `fsolve`, pour chaque valeur θ de la liste `Theta30`, les valeurs de θ_{10} et θ_{20} et remplir les listes `Theta10` et `Theta20`. S'appuyer sur l'aide ci-dessus.
-  Tracer la courbe donnant θ_{30} en fonction de θ_{10} . Ne pas oublier d'ajouter la légende et les axes.

```
1 ## Parametres
2 """ Theorie """
3 # Creation de la liste des valeurs pour Theta30
4 # Type array, allant de -90 a 0deg avec 91 points
5 # Penser a convertir en randian avec la fonction deg2rad de numpy par exemple
6 Theta30=""-- A completer --""
```

```
1 ## FONCTIONS
2 """ Theorie: Systeme d'equation a resoudre """
3 def Sys(X,arg):
4     ""-- A completer --"" # affectation de t1,t2 en fonction de X
5     ""-- A completer --"" # affectation de t3 en fonction de arg
6     ""-- A completer --"" # affectation de f1
7     ""-- A completer --"" # affectation de f2
8     return [f1,f2]
```

```
1 ## PROGRAMME PRINCIPAL
2 """ Theorie """
3 # Resolution du systeme avec fsolve
4 ""-- A completer --""
```

```
1 """ Tracer des courbes """
2 plt.grid()
3 plt.figure('A completer')
4 # plt.plot(A completer) # Theorie
5 plt.title(r"A completer")
6 plt.xlabel(r"A completer")
7 plt.ylabel(r"A completer")
```

FIGURE 4 – θ_{30} en fonction de θ_{10} (rad)

La Figure 4 montre la courbe théorique attendue.

4 | Étude expérimentale

4.1 Setup expérimental

Les mesures sur le portail ne peuvent pas s'effectuer pour le moment. Me demander pour que je vous donne les mesures expérimentales.

Le protocole aurait été le suivant :

4.2 Protocole expérimental

- ⚙ Placer le portail en position fermée et relever les angles du vantail et du moteur par rapport au bâti.
- ⚙ Faire une première acquisition pour une ouverture complète du portail (environ 20s).
- ⚙ Sauvegarder les données.
- ⚙ Placer le portail en position ouverte et relever les angles du vantail et du moteur par rapport au bâti.
- ⚙ Faire une seconde acquisition pour une fermeture complète du portail (environ 20s) et sauvegarder les données.

4.3 Traitement des données

Question 6 : Dans le script **Portail Domotic – Etudiant.py**, dans la section Experimentation: Recupération des données expérimentales :

- 🔗 En vous aidant de ce qui a déjà été fait pour extraire les données Solidworks, **définir une fonction** `donneesExp` qui prend en argument `NomFichier` et qui renvoie les deux listes correspondants aux tensions des capteurs de position.

Penser à gérer les lignes qui ne sont pas des données.

```
1 """ Simulation: Recupération des données Solidworks """
2 def donneesSldW(NomFichier):
3     #initialisation des listes resultats pour theta10 et theta30
4     t1sldw=[]
5     t3sldw=[]
6     with open(NomFichier,'r') as fichier:
7         # on lit les 3 lignes d'entete sans rien en faire
8         for i in range(3):
9             fichier.readline()
```

```

10         # on enregistre les valeurs des colonnes dans deux listes
11         for ligne in fichier :
12             t1,t3=fichier.readline().rstrip('\n').split()
13             t1sldw.append(float(t1))
14             t3sldw.append(float(t3))
15         return t1sldw,t3sldw
16
17     """ Experimentation: Recuperation des donnees experimentales """
18     """-- A completer --"""

```

4.3.1 Détermination des caractéristiques capteurs

Actuellement, les données expérimentales sont des tensions, or il faut des angles. Les capteurs mesurent en effet des tensions, il faut connaître la **caractéristique du capteur** pour retrouver la grandeur physique.

Ici, il est donné que la caractéristique des capteurs de positions du portail **est une droite**; la relation est alors

$$\theta_{\text{exp}} = V_{\text{exp}} \times G_{\theta}$$

Où G_{θ} est appelé le gain du capteur.

Question 7 : Dans la section PROGRAMME PRINCIPAL puis Experimentation, **appeler la fonction** donne-sExp définie précédemment, de sorte à importer les résultats d’une des expérimentations dans deux nouvelles listes V10Exp et V30Exp.

```

1     """ Experimentation """
2     # Appel de la fonction donneesExp pour importer les donnees dans les listes V10Exp et
      V30Exp
3     V10Exp,V30Exp="""-- A completer --"""

```

Avant de déterminer les gains capteurs, il vaut mieux s’assurer que les tensions sont bien définies.

Question 8 : Pour cela, compléter la partie # Tracer des tensions capteurs dans une nouvelle figure, afin de tracer V10Exp et V30Exp sur le même graphe.

```

1     # Determination des caracteristiques capteurs
2     # Tracer des tensions capteurs dans une nouvelle figure
3     plt.figure('A completer')
4     # plt.plot(A completer)
5     # plt.plot(A completer)
6     plt.title(r"A completer")
7     plt.xlabel(r"A completer")
8     plt.ylabel(r"A completer")

```

Question 9 : A l’aide de graphes, déterminer s’il est préférable d’utiliser la mesure en ouverture ou en fermeture.

Question 10 : En faisant l’hypothèse que les gains capteurs sont identiques, choisir s’il est préférable de déterminer le gain du capteur du bras moteur ou du vantail.

Question 11 : Remplir dans les lignes suivantes les positions angulaires initiale et finale afin que le gain du capteur puisse être calculé.

```

1     # Valeurs de initiales et finales de l'angle vantail (en radian)
2     Theta30_initiale = """-- A completer --"""
3     Theta30_finale = """-- A completer --"""

```

Les lignes qui suivent permettent de calculer les angles expérimentaux à partir des tensions capteurs puis de les tracer dans une nouvelle figure. Les courbes attendues devraient ressembler à celles présentées sur la Figure 5.

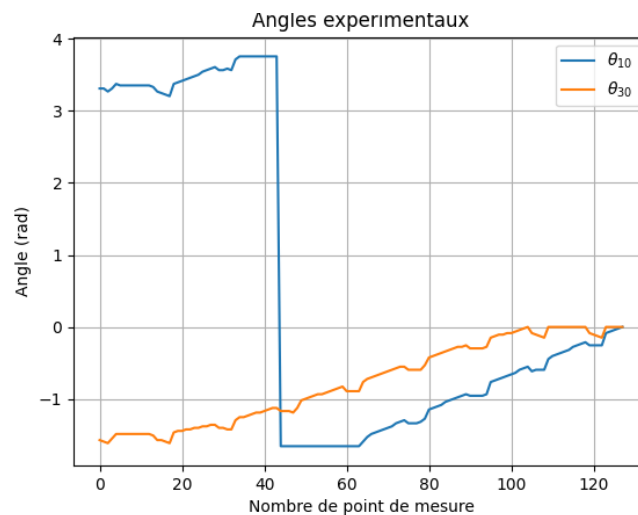


FIGURE 5 – Angles expérimentaux attendus

Question 12 : Enfin, tracer la courbe de θ_{30} en fonction de θ_{10} expérimentale sur le même graphique que la courbe théorique.

```
1 """ Tracer des courbes """
2 plt.grid()
3 plt.figure('A completer')
4 # plt.plot(A completer) # Theorie
5 plt.title(r"A completer")
6 plt.xlabel(r"A completer")
7 plt.ylabel(r"A completer")
8 # plt.plot(A completer) # Experimentation
```

5 | Étude numérique

5.1 Assemblage

-  Ouvrir le fichier **Portail Etudiant.sldasm**. La présentation des pièces est montrée en Figure 6.
-  Réaliser l'assemblage afin d'avoir 3 pivots et 1 pivot glissant.

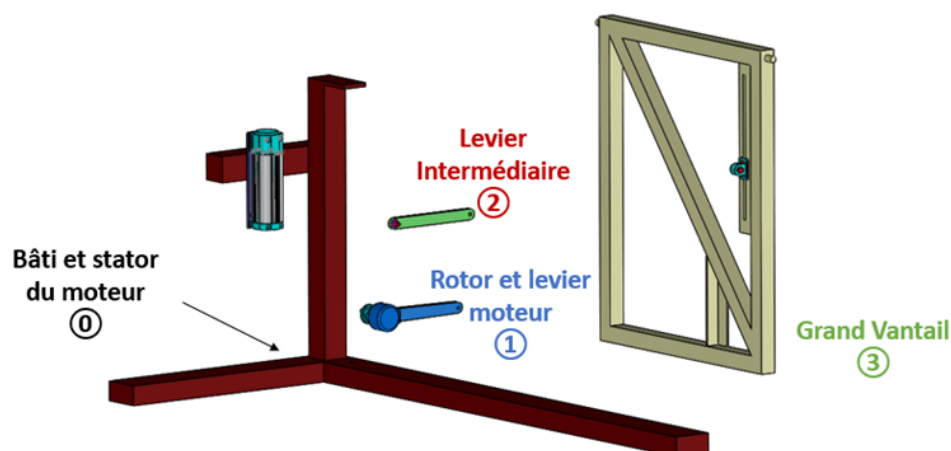


FIGURE 6 – Présentation des pièces de l'assemblage Solidworks

5.2 Animation

-  Si ce n'est pas déjà le cas, activer Meca3D.

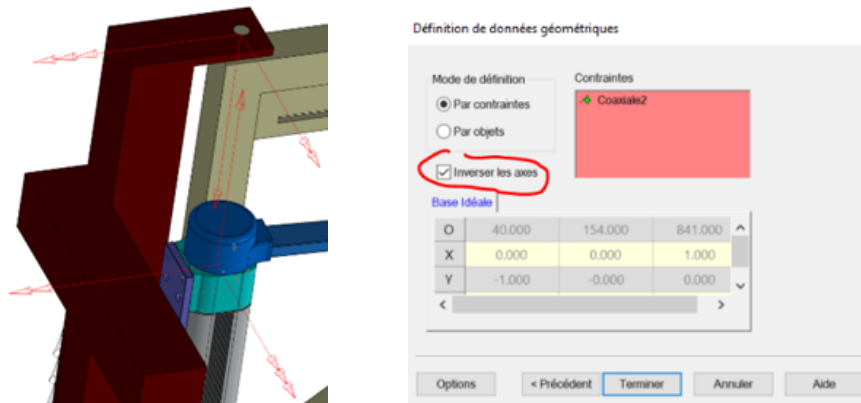


FIGURE 7 – Changer le sens d’une liaison pivot

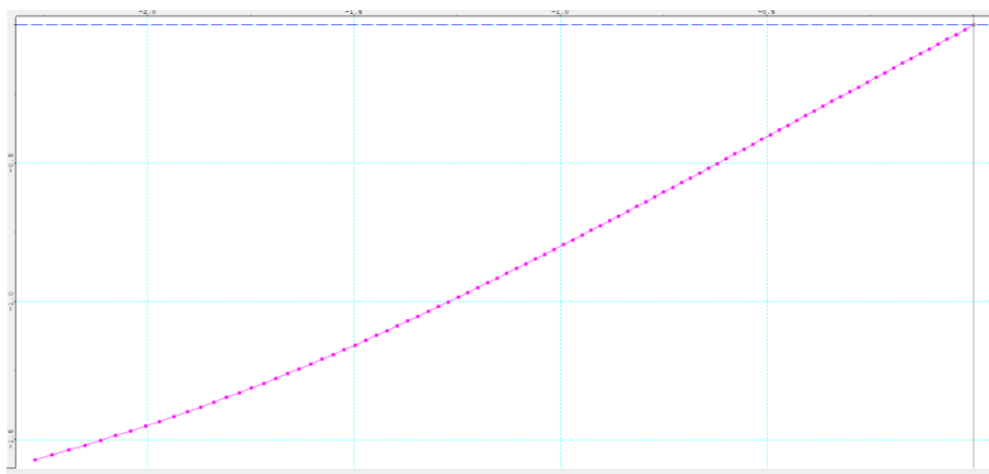


FIGURE 8 – Courbe obtenue sur Meca3D

- Sous Meca3D, faire un clic droit sur **Mécanisme** puis **Construction automatique**.
- Si les contraintes sont bonnes (attention aux arêtes!), Meca3D doit trouver les 4 liaisons souhaitée.

Il faut vérifier les axes des liaisons et si nécessaire les mettre dans le bon sens. La Figure 7 montre que les axes de la liaison Bâti-Grand vantail sont inversés. Pour les retourner :

- Faire un clic droit sur la liaison puis **Modifier** et **Suivant** deux fois.
- Cliquer sur **Inverser les axes** puis **Terminer**.
- Changer la position initiale du grand vantail vers la position fermé. *Par exemple en ajoutant une contrainte de parallélisme.*
- Réaliser le **Calcul mécanique**. Pour cela faire une **étude géométrique** en faisant tourner le grand vantail (la bonne liaison est à déterminer) de 0° à -90° et en prenant 91 positions.
- Lancer le calcul et vérifier le bon fonctionnement via l’animation.

5.3 Tracer de la courbe θ_{30} en fonction de θ_{10}

- Cliquez droit sur **Courbe** puis ajouter une courbe paramétrée.
- En utilisant les liaisons, tracer la courbe θ_{30} en fonction de θ_{10} .
- La courbe à obtenir est présentée sur la Figure 8.
- Exporter les données.

Question 13 : Dans le script **Portail Domotic – Etudiant.py** :

- Dans le programme principal, section **Simulation**, appeler la fonction `donneesS1dW` afin extraire les données issues de Solidworks.

 Tracer la courbe de θ_{30} en fonction de θ_{10} expérimentale sur le même graphique que la théorique et l'expérimentale.

```

1  """ Simulation """
2  # Appel de la fonction donneesSldW pour importer les donnees dans les listes Theta10SLDW
   et Theta30SLDW
3  """-- A completer --"""

1  """ Tracer des courbes """
2  plt.grid()
3  plt.figure('A completer')
4  # plt.plot(A completer) # Theorie
5  plt.title(r"A completer")
6  plt.xlabel(r"A completer")
7  plt.ylabel(r"A completer")
8  # plt.plot(A completer) # Experimentation
9  # plt.plot(A completer) # Simulation
10 plt.legend()
11 plt.show()

```

Annexe : étude théorique

Vous avez établi un système d'équation lors de l'étude théorique. On peut isoler θ_{20} en le mettant sous la forme :

$$\begin{cases} c \cos \theta_{20} = F(\theta_{10}, \theta_{30}) \\ c \sin \theta_{20} = G(\theta_{10}, \theta_{30}) \end{cases}$$

On élimine θ_{20} en élevant les lignes au carré puis en les additionnant. On peut alors obtenir une équation de la forme :

$$A(\theta_{10}) \cos(\theta_{30}) + B(\theta_{10}) \sin(\theta_{30}) = C(\theta_{10})$$

Où A , B et C sont des fonctions de θ_{10} .

Il y a plusieurs méthodes mais la plus élégante est de diviser l'équation par $\sqrt{A^2 + B^2}$ soit :

$$\frac{A}{\sqrt{A^2 + B^2}} \cos(\theta_{30}) + \frac{B}{\sqrt{A^2 + B^2}} \sin(\theta_{30}) = \frac{C}{\sqrt{A^2 + B^2}}$$

On peut alors poser ψ tel que :

$$\cos \psi = \frac{A}{\sqrt{A^2 + B^2}}; \sin \psi = \frac{B}{\sqrt{A^2 + B^2}}$$

L'équation s'écrit maintenant

$$\cos \psi \cos(\theta_{30}) + \sin \psi \sin(\theta_{30}) = \frac{C}{\sqrt{A^2 + B^2}}$$

D'après les formules de trigonométries, cela revient à

$$\cos(\theta_{30} - \psi) = \frac{C}{\sqrt{A^2 + B^2}}$$

Finalement

$$\theta_{30} = \psi \pm \arccos \frac{C}{\sqrt{A^2 + B^2}}$$

Avec une réflexion sur le signe de l'arccos.