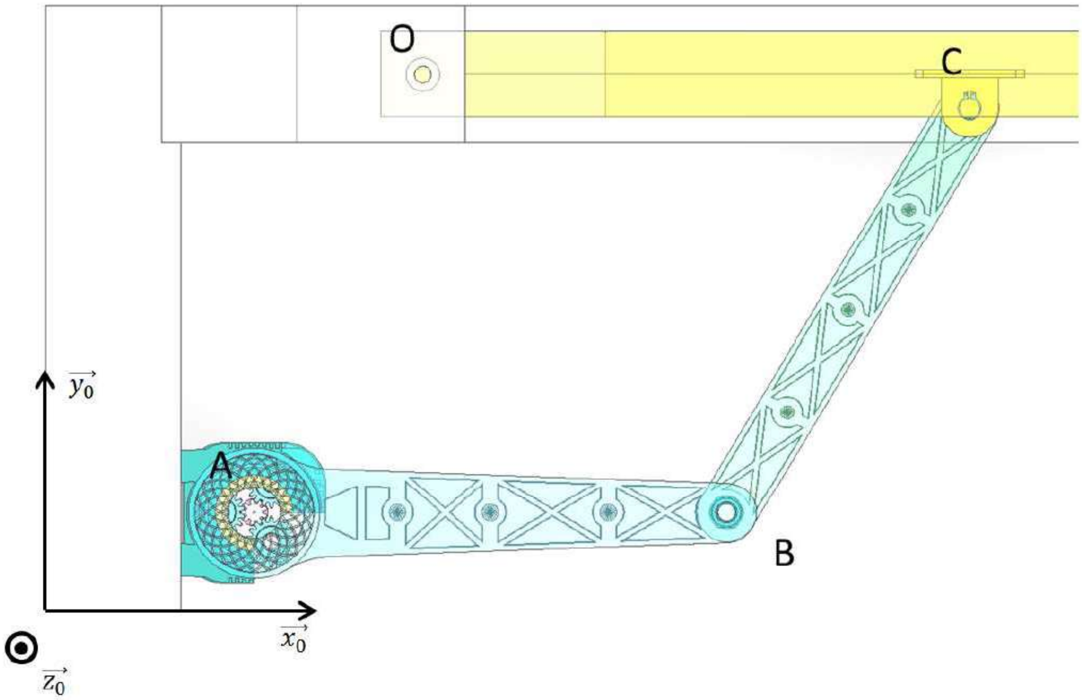
Informatique - TP 12

SI : bras articulés

Le but de ce TD est d'étudier un système de bras automatisés pour l'ouverture de vantaux.



On montre que l'angle (de sortie) θ_v d'ouverture du vantail et l'angle (d'entrée) θ_m du bras moteur vérifient la relation :

$$A + B\cos\theta_v + C\sin\theta_v - D\cos\theta_m - E\sin\theta_m - F\cos(\theta_v - \theta_m) - G\sin(\theta_v - \theta_m) = 0$$

où $A = 1954$, $B = 600$, $C = 1790$, $D = 560$, $E = 1400$, $F = 1960$ et $G = 112$ sont des longueurs constantes.
L'objectif de ce TD est d'obtenir à l'aide de méthodes numériques l'angle θ_v connaissant l'angle θ_m.
Pour cela, nous utiliserons la ***méthode de Newton*** qui permet d'obtenir une approximation des racines d'une équation du type $f(x) = 0$.
La méthode de Newton est la suivante :

- on fixe une valeur x_0 pour laquelle $f(x_0) \neq 0$ (sinon on dispose déjà d'une racine de l'équation !) ;
- on calcule les termes successifs de la suite définie par $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$;
- si cette suite converge, alors elle converge vers une racine de f.

1. Définir les constantes Python A, B, C, D, E, F, G données plus haut.
2. Définir une fonction `f(theta_m,theta_v)` qui à partir des valeurs θ_m et θ_v données en paramètres calcule et renvoie la valeur de $f(\theta_m, \theta_v) = A + B\cos\theta_v + C\sin\theta_v - D\cos\theta_m - E\sin\theta_m - F\cos(\theta_v - \theta_m) - G\sin(\theta_v - \theta_m)$.
3. Définir une fonction `df(theta_m,theta_v)` qui à partir des valeurs θ_m et θ_v données en paramètres calcule et renvoie la valeur de $\dfrac{\partial f}{\partial \theta_v}$.
4. Définir une fonction `newton(theta_m,epsilon=1e-5)` qui obtient, pour θ_m donné en paramètre, la valeur de θ_v correspondante en utilisant l'algorithme de Newton donné plus haut.

On prendra $x_0 = 0$ comme valeur de départ pour la suite et on poursuivra le calcul des valeurs de la suite tant que $|x_{n+1} - x_n| > \epsilon$ où ϵ est le paramètre nommé optionnel de la fonction `newton`.

5. Les valeurs de θ_m évoluent dans l'intervalle $[-2,4;0,2]$.
   Utiliser la fonction `newton` pour calculer les valeurs de θ_v correspondantes et tracer la représentation graphique de θ_v en fonction de θ_m.
   Animation de l'ouverture du vantail
   On souhaite obtenir une animation de l'ouverture du vantail.
   Pour obtenir une animation :
   - on fait une première représentation graphique, par exemple ici
     `vantail ,= plt.plot([0,np.cos(ThV[0])],[0,np.sin(ThV[0])])`
     représentera le vantail à l'instant initial sous la forme d'un segment ;
   - à intervalle de temps régulier (par exemple, tous les $\frac{1}{25}^{\text{ème}}$ de seconde), on met à jour cette représentation graphique.
     Pour s'assurer que les intervalles de temps sont bien réguliers, on utilise `plt.pause(dt)` où `dt` est la durée de la pause voulue en seconde.
     Pour mettre à jour le graphique, on utilise
     `vantail.set_data([0,np.cos(angle)],[0,np.sin(angle)])`
     à la place de `plt.plot`

   En supposant que θ_m va de 0 à $\dfrac{-\pi}{2}$ à une vitesse angulaire constante de $-1\texttt{rad.s}^{-1}$, s'arrête une seconde, puis revient à 0 avec une vitesse angulaire opposée, faire une animation de l'ouverture puis de la fermeture du vantail.