

Détermination des lois de mouvement – 60 minutes

Objectifs

- **B3-01** Vérifier la cohérence du modèle choisi en confrontant les résultats analytiques et/ou numériques aux résultats expérimentaux.
- **C1-04** Proposer une démarche permettant d'obtenir une loi entrée-sortie géométrique.
- **C2-06** Déterminer les relations entre les grandeurs géométriques ou cinématiques.
- **C3-01** Mener une simulation numérique.
- **D2-04** Choisir la grandeur physique à mesurer ou justifier son choix.
- **D2-05** Choisir les entrées à imposer et les sorties pour identifier un modèle de comportement.
- **A4-03** Interpréter et vérifier la cohérence des résultats obtenus expérimentalement, analytiquement ou numériquement.
- **A4-04** Rechercher et proposer des causes aux écarts constatés.

Objectifs

- Analyser les stratégies de ralliement avec trapèze.
- Déterminer les lois de commandes de chacun des axes pour ces stratégies de ralliement.
- Comparer la commande calculée avec la commande proposée par le logiciel.
- Comparer la commande avec le déplacement réel du bras beta.

Résoudre analytiquement

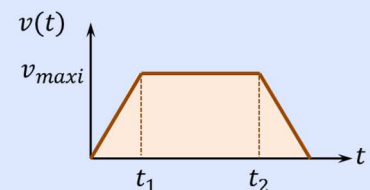
Activité 1. Lois géométriques

- Réaliser un schéma cinématique minimal paramétré du Control X.
- A partir de la documentation, donner la vitesse maximale de l'axe en translation [ms^{-1}] et la vitesse maximale du moteur en rotation [rad.s^{-1}].
- Donner également l'accélération maximale de l'axe en translation [ms^{-2}] et l'accélération maximale du moteur en rotation [rad.s^{-2}].

Résoudre analytiquement et numériquement

Activité 2. Lois de mouvement sur l'axe

- Ecrire la fonction `calcule_temps(amax :float, vmax :float, distance :float) -> float, float, float`, calculant les temps t_1 , t_2 et t_3 dans le cas où le mouvement est régi par un trapèze de vitesse dont l'accélération maximale est a_{max} , la vitesse maximale accessible est v_{max} , la distance à parcourir est distance .
- Ecrire une fonction `calcule_profil(amax, vmax, angle, dt) -> np.array, np.array, np.array, np.array` retournant :
 - `les_t` : tableau numpy des temps discrétisés toutes les dt s ;
 - `les_x` : tableau numpy des positions (en fonction du temps);
 - `les_v` : tableau numpy des vitesses (en fonction du temps);
 - `les_a` : tableau numpy des accélérations (en fonction du temps).
- Tracer les profils de position, vitesse et accélération de l'axe de translation, pour un déplacement de 100 mm.



Résoudre
analytiquement et
numériquement

Activité 3. Lois de mouvement du moteur

- Ecrire une fonction `calcule_moteur (amax, vmax, angle, dt)` -> `np.array`, `np.array`, `np.array`, `np.array` retournant :
 - `les_t` : tableau numpy des temps discrétisés toutes les `dt` s ;
 - `les_xr` : tableau numpy des positions angulaires (en fonction du temps);
 - `les_vr` : tableau numpy des vitesses angulaires (en fonction du temps);
 - `les_ar` : tableau numpy des accélérations angulaires (en fonction du temps).
- Tracer les profils de position, vitesse et accélération de l'axe du moteur, pour un déplacement du chariot de 100 mm.

Analyser les
écarts

Activité 4.

- Réaliser un essai en essai en trapèze pour un déplacement de 200 mm. (Analyse temporelle ► Trapèze de vitesse).
- Comparer les résultats issus de la modélisation et ceux issus de la simulation.

Synthèse

- **Réaliser une synthèse dans le but d'une préparation orale :**
 - Comparer les lois de commande pour un mouvement programmé avec les déplacements mesurés.
- 📁 Pour XENS – CCINP – Centrale – CCMP :
 - Donner l'objectif de l'activités.
 - Donner un schéma cinématique **en couleur et le paramétrage associé.**
- 📁 Pour CCMP :
 - Ajouter les courbes et les conclusions au compte-rendu.