

TP 4 : Modélisation dynamique d'un robot plan (2DDL).

1. Objectifs

L'objectif de ce travail pratique est d'aborder la modélisation dynamique d'un robot manipulateur. À l'issue de cette séance, l'étudiant doit être capable de :

- Comprendre la structure de l'équation dynamique d'un robot (modèle Lagrange-Euler).
- Calculer et interpréter les différentes matrices du modèle dynamique (matrice d'inertie, termes de Coriolis/centrifuges, gravité).
- Mettre en œuvre une simulation dynamique directe (calcul du mouvement à partir des couples).
- Implémenter et analyser un correcteur couple calculé (commande dynamique).

2. Manipulations

2.1. Définition du robot

On considère un robot plan à 2 articulations rotoïdes (RR) évoluant dans le plan vertical (x, y) - **avec prise en compte de la gravité**. Ce robot modélise un bras simple type "pelvis".

- Longueurs : $L1 = 0.5 \text{ m}$, $L2 = 0.4 \text{ m}$
- Masses : $m1 = 2 \text{ kg}$, $m2 = 1.5 \text{ kg}$ (considérées ponctuelles à l'extrémité de chaque segment)
- Inerties : On néglige les inerties tensorielle (segments considérés comme des masses ponctuelles)
- Gravité : $g = [0; -9.81]$ (agissant selon $-y$)

1. Création du robot avec paramètres dynamiques :

Dans la Robotics Toolbox, la fonction Link permet de spécifier les paramètres dynamiques. Le constructeur complet est :

Pour spécifier la dynamique, on utilise ensuite les propriétés :

- m : masse du lien
- r : position du centre de masse dans le repère du lien $[x, y, z]$
- I : matrice d'inertie (3x3) au centre de masse (négligée ici -> zeros(3,3))

- B : coefficient de frottement visqueux (négligé -> 0)
 - Jm : inertie du rotor (négligée -> 0)
2. **Exploration** : Affichez les informations du robot avec `robot.plot([0 0])` et `robot.teach()`. Observez l'effet de la gravité (le bras tombe si aucun couple n'est appliqué).

2.2. Structure du modèle dynamique

L'équation dynamique d'un robot manipulateur s'écrit sous la forme générale :

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + F(\dot{q}) = \tau$$

où :

- $M(q)$: matrice d'inertie (2x2) symétrique définie positive
- $C(q, \dot{q}) \dot{q}$: termes de Coriolis et centrifuges
- $G(q)$: couple dû à la gravité
- $F(\dot{q})$: frottements (négligés ici)
- τ : couples appliqués aux articulations

1. Calcul des matrices dynamiques :

Pour une configuration $q = [\pi/3, -\pi/6]$ et des vitesses $\dot{q} = [0.5, 0.2]$, calculez les différents termes

2. Analyse de la matrice d'inertie :

- Affichez M . Que remarquez-vous sur sa structure ? Pourquoi est-elle symétrique ?
- Calculez le déterminant de M pour différentes configurations. Est-il toujours positif ? Que se passe-t-il si $q_2 = 0$ ou $q_2 = \pi$?

3. Analyse des termes de Coriolis :

- Que se passe-t-il si $\dot{q} = [0, 0]$? Pourquoi ?
- Faites varier $\dot{q}_2$ en maintenant $\dot{q}_1$ constant et observez l'évolution de C .

2.3. Simulation dynamique directe

La simulation dynamique directe consiste à calculer l'évolution du système $(q(t), \dot{q}(t))$ à partir des couples appliqués $\tau(t)$.

1. Fonction d'état :

Écrivez une fonction `xdot = dyn_robot(t, x, robot, tau_func)` où :

- $x = [q_1; q_2; \dot{q}_1; \dot{q}_2]$ est l'état du système
- `tau_func(t, x)` est une fonction qui retourne les couples appliqués à l'instant t
- La fonction retourne $xdot = [\dot{q}_1; \dot{q}_2; \ddot{q}_1; \ddot{q}_2]$ avec $\ddot{q} = M(q)^{-1} * (\tau - C(q, \dot{q}) * \dot{q} - G(q))$

2. Simulation en chute libre :

- Simulez le comportement du robot lâché sans couple ($\tau = [0, 0]$) depuis une configuration initiale $q_0 = [\pi/2, 0]$ avec vitesse nulle.
- Tracez l'évolution des angles $q_1(t)$ et $q_2(t)$. Interprétez physiquement.
- Visualisez l'animation du mouvement avec `robot.plot(x(:,1:2))`.

3. Simulation avec couple constant :

- Appliquez un couple constant $\tau = [5, 2]$ (en Nm) et simulez le mouvement. Observez la différence.

2.4. Commande par couple calculé

La commande par couple calculé (computed torque control) est une loi de commande non linéaire qui linéarise et découple le système.

1. Principe :

La loi de commande s'écrit :

$$\tau = M(q) u + C(q, \dot{q}) \dot{q} + G(q)$$

où u est une nouvelle commande (généralement un correcteur linéaire). En boucle fermée, le système devient :

$$\ddot{q} = u$$

soit un système linéaire découplé (double intégrateur).

2. Choix de la commande linéaire :

Choisissons u comme un correcteur PD avec anticipation de vitesse et accélération :

$$u = \ddot{q}_d + K_v (\dot{q}_d - \dot{q}) + K_p (q_d - q)$$

où $q_d(t), \dot{q}_d(t), \ddot{q}_d(t)$ est une trajectoire de référence.

3. Implémentation :

- Générez une trajectoire de référence pour les articulations. Par exemple, un mouvement polynomial du repos ($q_d(0) = [0,0]$, $q_d(T) = [\pi/2, \pi/3]$) avec des vitesses nulles au début et à la fin (polynôme d'ordre 3 ou 5).
- Implémentez la boucle de simulation avec la commande par couple calculé.
- Simulez et comparez la trajectoire réelle à la trajectoire désirée. Tracez l'évolution des erreurs.

4. Analyse de robustesse :

- Que se passe-t-il si les paramètres dynamiques (masses, longueurs) sont mal estimés ? Simulez avec une erreur de 20% sur m_1 et m_2 .
- Ajoutez des frottements non modélisés dans le robot simulé (en modifiant `dyn_robot`) mais pas dans le modèle de commande. Observez l'erreur statique.

2.5. Identification des paramètres dynamiques (optionnel)

La toolbox permet de générer des trajectoires d'excitation pour l'identification des paramètres dynamiques.

1. **Principe** : L'équation dynamique est linéaire par rapport aux paramètres dynamiques (masses, centres de masse, inerties) :

$$\tau = \Phi(q, \dot{q}, \ddot{q}) \pi$$

où π est le vecteur des paramètres à identifier.

2. **Génération de trajectoire**

Ecrire le programme permettant de générer une trajectoire.

4. Compte-rendu

Le compte-rendu devra contenir :

1. **Introduction** : Description du robot et des objectifs du TP.
2. **Modélisation dynamique** :
 - Analyse de la structure des matrices M , C , G .
 - Commentaires sur leurs propriétés (symétrie, définie positivité, etc.).
3. **Simulation dynamique** :
 - Résultats de la simulation en chute libre et avec couples constants.
 - Analyse physique du comportement observé.
4. **Commande par couple calculé** :
 - Implémentation de la loi de commande.
 - Courbes d'évolution des erreurs de suivi de trajectoire.
 - Analyse de robustesse face aux incertitudes paramétriques.
5. **Discussion** :
 - Avantages et limites de la commande par couple calculé.
 - Problèmes liés à la connaissance exacte du modèle.
 - Pistes d'amélioration (commande adaptative, robuste).
6. **Conclusion** : Synthèse des compétences acquises.
7. **Annexes** : Codes MATLAB commentés.

Les fichiers source Matlab (.m) devront être joints au compte-rendu.