

TP 6 : Commande dynamique d'un robot

1. Objectifs

L'objectif de ce travail pratique est d'implémenter et d'analyser différentes lois de commande dynamique pour un robot manipulateur. À l'issue de cette séance, l'étudiant doit être capable de :

- Implémenter une commande PID (indépendante par articulation) et analyser ses performances.
- Implémenter une commande par couple calculé (computed torque control) et comprendre ses avantages.
- Implémenter une commande prédictive (MPC) simple.
- Comparer les performances des différentes lois de commande (poursuite de trajectoire, rejet de perturbations, robustesse).
- Analyser l'influence des gains et des incertitudes paramétriques.

2. Manipulations

2.1. Définition du robot et du modèle de simulation

1. On reprend le robot plan vertical à 2 degrés de liberté du TP4.
2. **Fonction de simulation** : On réutilise la fonction `dyn_robot` développée dans le TP4 pour simuler la dynamique du robot.

2.2. Génération de trajectoire de référence

Pour tester les lois de commande, on génère une trajectoire de référence articulaire (polynôme quintique) entre deux configurations.

2.3. Commande PID indépendante par articulation

La commande PID est la plus simple et la plus répandue dans l'industrie. Chaque articulation est contrôlée indépendamment comme si elle était découplée des autres.

2.3.1. Structure du correcteur

La loi de commande PID s'écrit :

$$\tau_i = K_{p_i} * (q_{d_i} - q_i) + K_{d_i} * (\dot{q}_{d_i} - \dot{q}_i) + K_{i_i} * \int (q_{d_i} - q_i) dt$$

2.3.2. Simulation et analyse

1. **Simulation** : Lancez la simulation avec le correcteur PID.
2. **Analyse des performances** :
 - Tracez l'évolution des angles réels superposés aux angles de référence.
 - Tracez l'évolution des erreurs de poursuite.
 - Calculez l'erreur quadratique moyenne (RMSE) pour chaque articulation.
3. **Influence des gains** :
 - Augmentez/diminuez les gains proportionnels K_p . Observez l'effet sur la rapidité et le dépassement.
 - Augmentez/diminuez les gains dérivés K_d . Observez l'effet sur l'amortissement.
 - Ajoutez l'action intégrale K_i et observez l'effet sur l'erreur statique.
4. **Couplage** : Observez l'effet d'une consigne sur l'articulation 1 sur l'articulation 2 (et inversement). Le PID parvient-il à découpler naturellement le système ?

2.4. Commande par couple calculé

La commande par couple calculé (computed torque control) utilise le modèle dynamique pour linéariser et découpler le système.

2.4.1. Structure du correcteur

Rappel de la loi de commande vue au TP4 :

$$\begin{aligned}\tau &= M(q) u + C(q, \dot{q}) \dot{q} + G(q) \\ u &= \ddot{q}_d + K_v (\dot{q}_d - \dot{q}) + K_p (q_d - q)\end{aligned}$$

2.4.2. Simulation et analyse

1. **Simulation** : Lancez la simulation avec le correcteur CTC.
2. **Comparaison PID vs CTC** :
 - Comparez les erreurs de poursuite sur un même graphique.
 - Comparez les couples appliqués (saturations éventuelles).
 - Que peut-on conclure sur le découplage ?
3. **Analyse de la dynamique d'erreur** :
 - Théoriquement, l'erreur doit suivre : $\ddot{e} + K_v \dot{e} + K_p e = 0$.
 - Vérifiez cette équation à partir des résultats de simulation.
 - Choisissez K_p et K_v pour obtenir un comportement critique ($\zeta=1$) avec un temps de réponse désiré.

2.4.3. Robustesse

1. **Incertitudes paramétriques** : Modifiez les paramètres dynamiques dans le modèle de commande (pas dans le modèle de simulation).

2. Analyse :

- Re-simulez avec le modèle erroné dans la loi de commande.
- Comparez les performances avec le cas nominal.
- Jusqu'à quelle erreur paramétrique le système reste-t-il stable ?

2.5. Commande prédictive (MPC) simple

La commande prédictive utilise un modèle du système pour prédire son comportement futur et optimiser la commande sur un horizon.

2.5.1. Principe simplifié

Pour un système échantillonné, à chaque instant k , on résout le problème d'optimisation :

$$\min_U \sum_{j=1}^N \|q(k+j|k) - q_{ref}(k+j)\|_Q^2 + \|\tau(k+j-1)\|_R^2$$

sous contraintes dynamiques et éventuellement contraintes sur les états/commandes.

2.5.2. Implémentation simplifiée (linéarisation)

Pour éviter la complexité d'un MPC non linéaire, on linéarise le système autour de la trajectoire de référence.

2.5.3. Utilisation de l'MPC Toolbox (optionnel)

Si l'MPC Toolbox est disponible, on peut l'utiliser pour une implémentation plus rigoureuse.

2.6. Analyse comparative approfondie

2.6.1. Métriques de performance

Pour chaque loi de commande, calculez :

1. **Erreur de poursuite** : RMSE sur chaque articulation et sur la position cartésienne.
2. **Énergie de commande** : $\int \|\tau(t)\|^2 dt$.
3. **Temps de calcul** : temps CPU moyen par pas de simulation.
4. **Robustesse** : sensibilité aux incertitudes paramétriques (faites varier les erreurs de 0% à 50% et observez la dégradation des performances).

2.6.2. Tableau comparatif

Critère	PID	CTC	MPC
Erreur de poursuite (RMSE)			
Énergie de commande			
Rejet de perturbation			
Robustesse paramétrique			

2.6.3. Test de perturbation

Ajoutez une perturbation impulsionnelle (choc) sur le robot à $t = 2s$. Comparez la capacité de chaque correcteur à rejeter cette perturbation.

2.6.4. Respect des contraintes

Ajoutez des contraintes sur les couples (saturation). Quel correcteur gère le mieux la saturation ? Lequel est le plus affecté ?

2.7. Extension : Commande adaptative (optionnel)

1. **Principe** : La commande adaptative estime les paramètres dynamiques en ligne et ajuste la loi de commande.
2. **Test** : Simulez avec des paramètres initiaux erronés et observez la convergence des estimations.

3. Compte-rendu

Le compte-rendu devra contenir :

1. **Introduction** : Objectifs du TP et rappel du système étudié.
2. **Méthodologie** :
 - Description des différentes lois de commande implémentées.
 - Explication du réglage des gains.
3. **Résultats expérimentaux** :
 - Graphiques comparatifs des erreurs de poursuite pour chaque correcteur.
 - Analyse de l'influence des gains (PID) et du réglage de la dynamique d'erreur (CTC).
 - Étude de robustesse face aux incertitudes paramétriques.
 - Réponse aux perturbations.
 - Tableau comparatif synthétique.
4. **Discussion** :
 - Avantages et inconvénients de chaque approche.
 - Compromis performance/robustesse/complexité.
 - Choix d'une loi de commande selon l'application (robot industriel, cobot, robot spatial...).
5. **Conclusion** : Synthèse des compétences acquises et ouvertures (commandes robustes, adaptatives, intelligentes).
6. **Annexes** : Codes MATLAB commentés.

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