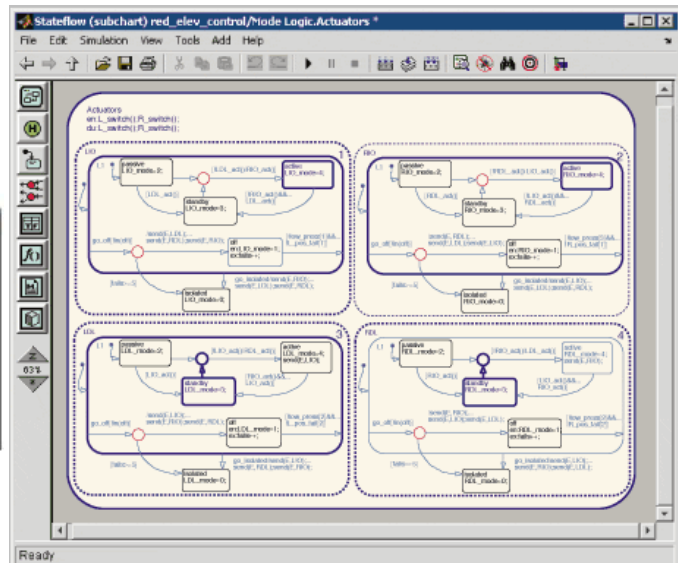
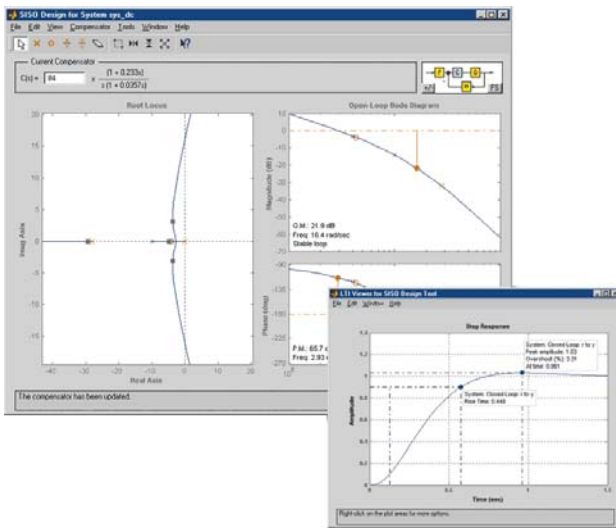
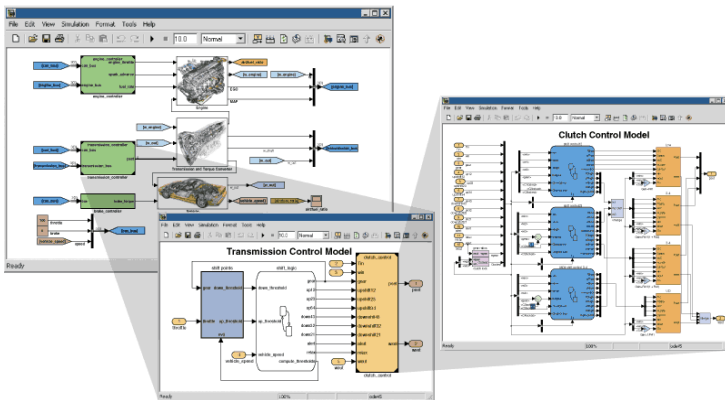
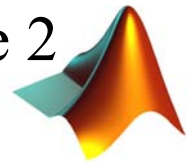


Simulation des systèmes continus

MATLAB[®] Séance 2



INSA de Lyon
3GI
2005-2006
Lounis ADOUANE

ETUDE DE SYSTEMES DYNAMIQUES

La trajectoire d'un outil, l'évolution de la température d'un four, le cours d'une action, le nombre de pièces produites sont autant d'exemples de systèmes dynamiques traités par le Génie Industriel (GI). À partir de conditions initiales particulières on a souvent besoin de déterminer l'évolution du système en fonction du temps, par exemple : le nombre de pièces produites va-t-il augmenter ? L'outil va-t-il suivre la trajectoire prévue, etc. Les systèmes étudiés en GI peuvent donc être autant discrets que continus, linéaires que non linéaires.

L'objectif premier de cette deuxième séance de TP est de vous faire explorer davantage les possibilités et les méthodes qu'offre MATLAB pour simuler des systèmes dynamiques continus. Les bras manipulateurs font partie de ces systèmes dynamiques continus et fortement non linéaires qu'un ingénieur en GI aura éventuellement l'occasion de manipuler lors de ses investigations académiques ou industrielles. Pour cela, ce TP se propose d'étudier le cas d'un système robotique avec un espace de travail¹ planaire.

Points qui seront traités :

- Programmation MATLAB avec des m-file (des scripts et des fonctions),
- Modélisation et contrôle des systèmes sous Simulink,
- Interfaçage entre des programmes m-file et les schémas-blocs Simulink,
- Utilisation des GUI's (Graphical User Interface),
- Notions d'automatique,
- Notions de robotique,
- Animation graphique des mouvements du robot.

I MODELISATION ET COMMANDE D'UN BRAS MANIPULATEUR PLANAIRE

Le robot SCARA (Selective Compliance Assembly Robot Arm) (Figure.1(a)) est l'un des robots les plus couramment utilisés en industrie. La version à deux degrés de liberté (2ddl) du SCARA va nous servir dans ce TP comme base d'étude pour le contrôle d'un système robotique. Afin de vous permettre de comprendre et de traiter efficacement ce TP, des définitions succinctes des principales notions en robotique vont vous être préalablement exposées. La figure 1 (b) donne une représentation graphique d'un robot SCARA à 2ddl.

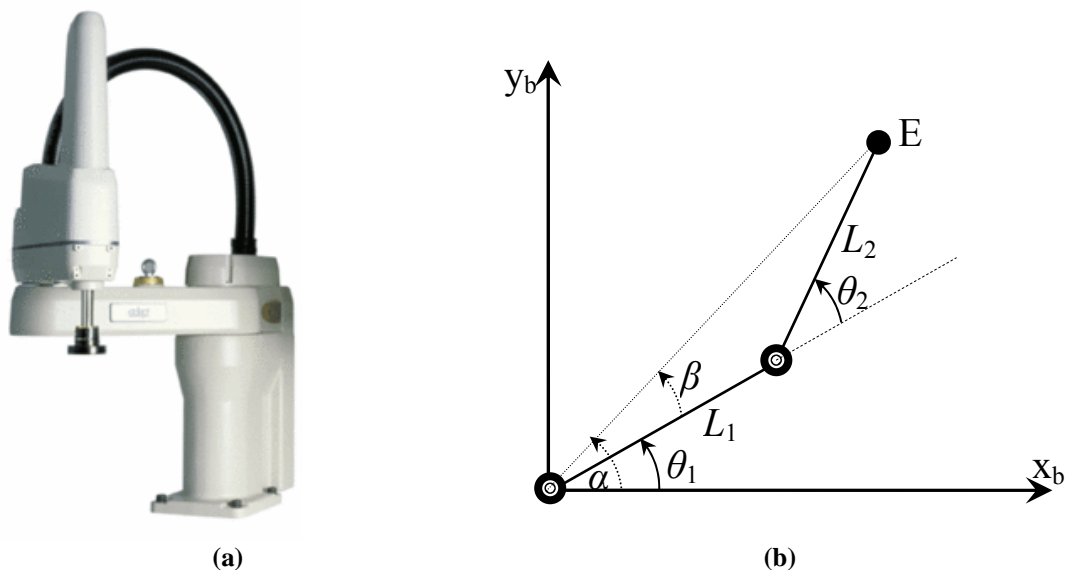


Figure. 1

¹ Domaine atteignable par l'effecteur du robot.

La configuration du robot est déterminée par les variables articulaires θ_1 et θ_2 . La position et l'orientation l'effecteur (E) du robot en coordonnées homogènes (Khalil 99)² sont données par :

$$T_e^b(q) = \begin{bmatrix} \vec{n}_e^b & \vec{s}_e^b & \vec{a}_e^b & \vec{p}_e^b \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \\ 0 & -\cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) & L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Avec : $\vec{s}_e^b, \vec{n}_e^b, \vec{a}_e^b$ désignent respectivement les vecteurs unitaires suivant les axes x_e, y_e et z_e du repère R_e exprimés dans le repère R_b et où le \vec{p}_e^b est le vecteur exprimant l'origine du repère R_e dans le repère R_b .

La conception et la commande des robots manipulateurs nécessitent le calcul de certains modèles mathématiques, parmi lesquels le :

Modèle géométrique direct (MGD) : le MGD est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, i.e., les coordonnées opérationnelles du robot X , en fonction de ses coordonnées articulaires ($q \equiv \theta$). Ceci s'exprime par une fonction f définie comme suit : $X = f(q)$.

Modèle géométrique inverse (MGI) : le MGI consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constitue ce que l'on appelle le MGI, $q = f^{-1}(X)$. Il est à noter qu'il y a plusieurs méthodes systématiques d'obtention du MGI (Khalil 99). Dans le cas du TP, les solutions du MGI sont obtenues via une méthode géométrique. Ceci est rendu possible vu la simplicité relative du problème traité.

Modèle Dynamique Inverse (MDI) : le MDI est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires. Le MDI est représenté par une relation de la forme : $\Gamma = f(q, \dot{q}, \ddot{q}, f_e)$

Avec :

- Γ vecteur des couples/forces des actionneurs, selon que l'articulation est rotoïde ou prismatique,
- q, \dot{q}, \ddot{q} correspondent respectivement aux positions, vitesses et accélérations articulaires,
- f_e vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement.

Modèle Dynamique Direct (MDD) : le MDD est celui qui exprime les accélérations articulaires en fonction des positions, vitesses et couples des articulations. C'est le modèle le plus usité en simulation car il se prête bien à l'intégration numérique, il est alors représenté par la relation : $\ddot{q} = f(q, \dot{q}, \Gamma, f_e)$.

Voir l'Annexe 1 pour plus de détails sur les équations exprimant la dynamique d'un robot manipulateur.

Pour pouvoir commander convenablement le mouvement d'un robot il est impératif de connaître aussi quelques définitions relatives à la génération de mouvement et à la commande de ces mouvements.

Génération de mouvement : le problème de la génération de mouvement est de calculer les consignes de référence en position, vitesse et accélération qui sont fonction du temps et qui assurent le passage du robot (son effecteur) par une trajectoire désirée, définie par une suite de situations de l'organe terminal ou de configuration articulaires.

Parmi les différentes classes de mouvements existantes on peut distinguer celles-ci :

1. un mouvement entre deux points avec une trajectoire libre entre les points,
2. un mouvement entre deux points avec une trajectoire contrainte entre les points (trajectoire rectiligne par exemple),

² Khalil 99, W. Khalil et E. Dombre, Modélisation, identification et commande des robots, Hermes 1999.

La première classe correspond généralement à la génération de mouvement directement dans l'espace articulaire. Dans le cas de la deuxième classe, la trajectoire étant décrite dans l'espace opérationnel (en fonction de (x,y,z)), il est préférable donc de raisonner directement dans cet espace.

Nous allons nous restreindre dans ce TP à la deuxième classe de mouvement. Les méthodes de passage d'une configuration articulaire q_i (ou opérationnel X_i) à une configuration q_f / X_f sont diverses, parmi elles il y a par exemple : l'interpolation polynomiale ou la loi Bang-Bang (utilisée pour minimiser le temps nécessaire pour atteindre la configuration finale du robot).

Commande des mouvements : Le modèle dynamique d'un robot décrit un système de n équations différentielles du second ordre non linéaires et couplées (voir Annexe 1), n étant le nombre d'articulations. Pourtant, dans une commande classique, qui est celle de la plupart des robots industriels, le système robotique est considéré comme un système linéaire et chacune de ses articulations est asservie par une commande décentralisée de type PID à gains constants. Ses avantages sont la facilité d'implantation et le faible coût de calcul. En contrepartie, la réponse temporelle du robot variant selon sa configuration, on constate des dépassements de consigne et une mauvaise précision de suivi dans les mouvements rapides. Dans beaucoup d'applications, ces inconvénients ne représentent pas un gros handicap.

Dans ce type de commande les actions du PID sont les suivantes :

$$\begin{cases} K_{pj} = 3a_j \omega_j^2 \\ K_{dj} = 3a_j \omega_j \\ K_{fj} = a_j \omega_j^3 \end{cases}$$

Avec :

- $a_j = A_{jj \max}$ désigne la valeur maximale de l'élément A_{jj} de la matrice d'inertie du robot,

- $\omega_j > 0$ et est choisi la plus grand possible, toutefois, cette pulsation ne devra pas être supérieure à la pulsation de résonance ω_{rj} du système mécanique afin de ne pas le déstabiliser.

Pour plus de détails sur la méthode d'obtention des actions du PID pour un robot manipulateur reportez-vous à l'Annexe 2.

II PROJET DE ROBOTIQUE SOUS MATLAB

L'objectif de ce TP est l'utilisation de MATLAB pour modéliser et commander un robot de type SCARA à deux degrés de liberté. A ces fins, un ensemble de programmes vous sont préalablement fournis³, et vous permettront de disposer des principaux éléments pour mener à bien votre TP. D'autre part, avant d'effectuer le moindre travail, récupérez et copiez sur votre poste de travail le répertoire contenant l'ensemble des programmes sur lesquels vous devez travailler. Les fichiers et leur contenu sont les suivants :

SimulationRobotSCARA.m C'est le programme principal, il permet en l'exécutant d'ouvrir une interface utilisateur (Figure. 2) qui vous permettra de simuler le système robotique soit en statique⁴ soit en dynamique. Une représentation graphique du robot SCARA 2ddl ainsi que la possibilité de l'animer sont accessibles via cette interface. Cette interface est aussi disposée pour récupérer les événements émanant de la souris. En l'occurrence, la souris va vous permettre dans le cadre de votre projet de TP de donner des consignes de position (x, y) à atteindre par l'organe terminal du robot.

ModeleGeometriqueDirect.m Définit les positions (x_i, y_i) des différentes articulations du SCARA en fonction de ses coordonnées articulaires

ModeleGeometriqueInverse.m Définit les coordonnées articulaires du SCARA en fonction de la position de son effecteur

ModeleDynamiqueDirect.m Définit l'accélération des articulations du SCARA en fonction des couples appliqués par ses actionneurs

TrajectoireConsigne.m Donne dans le domaine opérationnel la trajectoire consigne à faire suivre par l'effecteur du SCARA

GestionGUI.m Gère tous les événements (clic de souris, bouton enfoncé, etc.) affectant la fenêtre GUI (Graphical User Interface)

SetAffichage.m Permet de remettre à jour la représentation graphique du robot SCARA ainsi que l'affichage des informations articulaires et la position de l'effecteur (Figure. 2)

SimulinkControleRobotSCARA.mdl Modèle Simulink (Figure. 3) qui permet en l'interfaçant avec les programmes décrits ci-dessus de commander le mouvement du SCARA afin de suivre une trajectoire consigne.

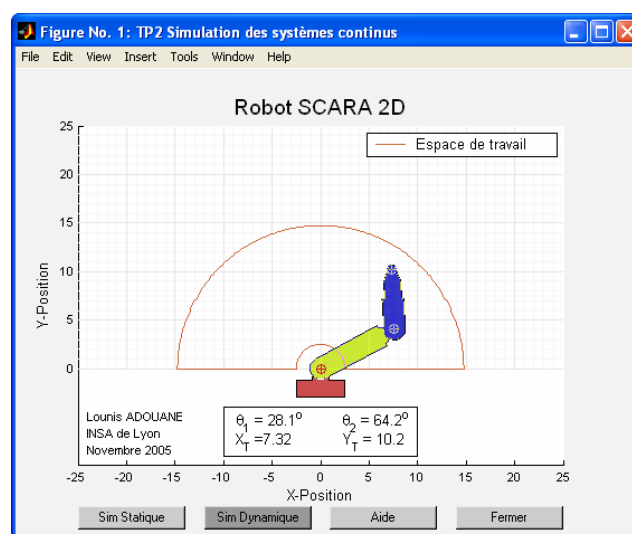


Figure. 2

³ \\gpr1serv\promo14\TP_MATLAB\ProgrammesSeance2

⁴ Sans l'introduction des paramètres inertiels, de frottements ou de perturbations affectant le robot.

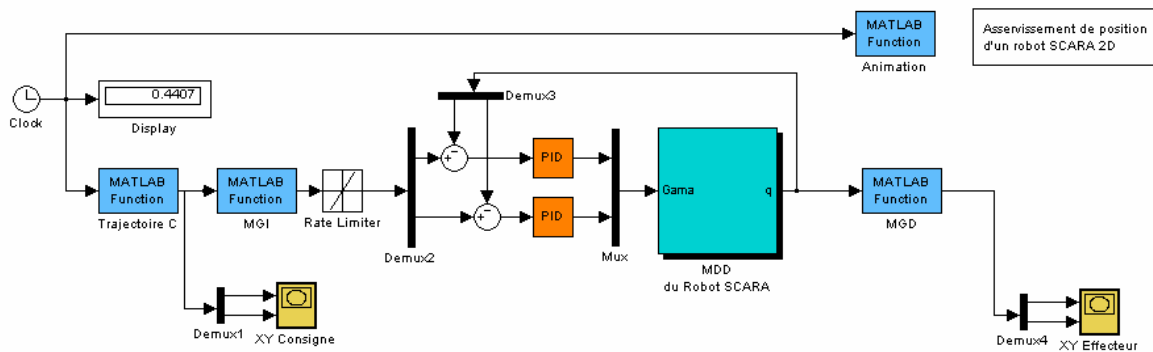


Figure. 3

II.1 Le travail à faire

Tester et explorer attentivement les programmes fournis ainsi que le modèle Simulink afin de comprendre les méthodes adoptées pour réaliser les éléments préalablement accessibles pour la simulation. Vous devez ajouter à ces programmes des parties de codes (pour les cas des m-files) ou des schémas-blocs (pour le cas des modèles Simulink) qui vont permettre soit d'améliorer les fonctionnalités du mini-logiciel fourni, soit d'ajouter de fonctionnalités complètement nouvelles. Il est à noter que, pour une meilleure lisibilité de vos programmes, il est fortement conseillé de les commenter.

Compréhension des programmes :

1. Quel est le rôle de la variable T ? Donnez la partie de code et les éléments de la simulation affectés par cette variable.
2. Dans le cas de la simulation du modèle dynamique du robot, quelle est la trajectoire consigne imposée à l'effecteur du robot pour relier sa position initiale à la position finale désirée ? Donnez la partie de code représentant cela.
3. Quelle est la condition pour que les événements de la souris soient pris en compte dans les programmes MATLAB ? Reproduisez et expliquez les lignes de programme responsables de cela.
4. Donnez et expliquez la partie de programme réalisant l'inversion de la matrice d'inertie du robot SCARA.
5. Expliquez le fonctionnement de la fonction *SetAffichage.m*.
6. Répertoriez les différents types de blocs (leurs noms et à quelle liste ils appartiennent) utilisés dans le modèle Simulink fourni. Expliquez pour chaque bloc le rôle qu'il occupe pour réaliser la commande du robot.

Programmation MATLAB/Simulink :

7. Affichez dans une même figure les caractéristiques liées à la commande du robot SCARA pour suivre la trajectoire référence. Les éléments qui doivent être visibles sur la figure sont :
 - les coordonnées articulaires consignes et effectives des articulations,
 - les positions (x,y) consignes et effectives de l'organe terminal,
 - l'erreur de suivi de trajectoire,
 - les couples articulaires.

Indication : utilisez le schéma-bloc «To Workspace» dans la liste «Sinks» de Simulink.

Remarque : cette partie de code doit apparaître dans le programme **GestionGUL.m** et plus précisément entre «%% Début code affichage figure %%» et «%% Fin code affichage figure %%».

8. Combien de solutions articulaires existe-t-il pour chaque configuration (x, y) de l'organe terminal d'un robot planaire ? Illustrez cela par un graphique. Introduisez cette constatation dans votre projet pour qu'aucune articulation du robot ne sorte de son espace de travail ?

9. Les paramètres actuels des PID (figure.3) ont été fixés empiriquement. Synthétisez les contrôleurs PID adaptés au robot SCARA en vous inspirant de l'Annexe 2. Une fois les paramètres des PID obtenus par programmation, vous devez automatiquement les affecter aux blocs correspondants. **Indication** : inspirez-vous de la définition et de l'utilisation de la variable globale « T », **Remarque** : cette partie de code doit apparaître dans le programme *SimulationRobotSCARA.m* et plus précisément entre « %% Début code PID %% » et « %% Fin code PID %% ».
10. Etudiez l'influence des actions Proportionnelle, Dérivée et Intégrale pour le suivi de trajectoire de l'effecteur du robot.
11. Générez une trajectoire référence de type carré et circulaire en vous assurant que l'effecteur du robot ne sort jamais de son espace de travail. Vérifiez à chaque fois que les caractéristiques de l'asservissement sont correctes.
12. Utilisez la commande MENU pour choisir d'une manière interactive la trajectoire consigne à donner au robot.
13. Avec un schéma-bloc de type « Switch » faites en sorte de pouvoir changer d'une manière interactive la consigne à suivre par le robot (ex. consigne en échelon ou en rampe, consignes programmées).
14. Rédigez l'aide correspondant à l'ensemble des fonctionnalités que vous avez programmées, cette Aide doit être affichée lors de l'appui sur le bouton poussoir «Aide» (Figure. 2).

ANNEXE

Annexe 1

Le formalisme de Lagrange est l'un des plus utilisés pour le calcul du modèle dynamique des robots. Ce formalisme décrit les équations du mouvement en terme de travail et d'énergie du système, ce qui se traduit, lorsque l'effort extérieur sur l'organe terminal est supposé nul, par l'équation (cas d'un robot à « n » ddl) :

$$\Gamma_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \quad i = 1, \dots, n$$

Avec :

- Γ_i : couple/force articulaire engendré par l'actionneur (articulation) i ,
- L : lagrangien du système égale à $E - U$,
- E : énergie cinétique totale du système,
- U : énergie potentielle totale du système.

Après simplification :

$$\Gamma = A(q) \cdot \ddot{q} + H(q, \dot{q})$$

Avec :

- A est la matrice ($n \times n$) de l'énergie cinétique, elle est appelée aussi matrice d'inertie du robot, elle est symétrique et définie positive et ses éléments sont fonction des variables articulaires q .
- H est un vecteur caractéristique dépendant à la fois de q et de \dot{q} (Khalil 99).

Les matrices A et le vecteur H sont fonction des paramètres géométriques et inertiels du robot. Les équations dynamiques d'un système mécanique articulé forment donc un système de n équations différentielles du second ordre, couplées et non linéaires.

Annexe 2

La loi de commande d'un PID est donné par :

$$\Gamma = K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) + K_I \int_{t_0}^t (q_d - q) dt$$

où $\dot{q}_d(t)$ et $q_d(t)$ désignent les vitesses et positions désirées dans l'espace articulaire et K_p , K_d et K_I sont des matrices diagonales définies positives, de dimension $(n \times n)$, d'éléments génériques respectivement les gains proportionnels K_{pj} , dérivés K_{dj} et intégraux K_{Ij} .

Le calcul des gains K_{pj} , K_{dj} et K_{Ij} est effectué en considérant le modèle de l'articulation j représenté par le système linéaire du deuxième ordre à coefficients constants suivant :

$$\Gamma_j = a_j \ddot{q}_j + F_{vj} \dot{q}_j + \gamma_j$$

Équation dans laquelle $a_j = A_{jj \max}$ désigne la valeur maximale de l'élément A_{jj} de la matrice d'inertie du robot, F_{vj} les frottements visqueux et γ_j représente un couple perturbateur.

En négligeant F_{vj} et γ_j la fonction de transfert du système en boucle fermée est donné par :

$$\frac{q_j(s)}{q_{jd}(s)} = \frac{K_{dj}s^2 + K_{pj}s + K_{Ij}}{a_j s^3 + K_{dj}s^2 + K_{pj}s + K_{Ij}}$$

et l'équation caractéristique s'écrit donc :

$$\Delta(s) = a_j s^3 + K_{dj}s^2 + K_{pj}s + K_{Ij}$$

La solution la plus courante en robotique consiste à choisir les gains de manière à obtenir un pôle triple réel et négatif, ce qui donne la réponse la plus rapide possible sans oscillation. Par conséquent, l'équation caractéristique se factorise de la façon suivante :

$$\Delta(s) = a_j (s + \omega_j)^3$$

avec $\omega_j > 0$, et est choisi la plus grand possible, toutefois, cette pulsation ne devra pas être supérieure à la pulsation de résonance ω_{rj} du système mécanique pour ne pas le déstabiliser.

$$\text{On en déduit finalement les gains : } \begin{cases} K_{pj} = 3a_j \omega_j^2 \\ K_{dj} = 3a_j \omega_j \\ K_{Ij} = a_j \omega_j^3 \end{cases} .$$