Commande Non-linéaire: TP 3

Ph. Müllhaupt

17-12-18

1 Bonus

Ce TP se présente comme une succession de problèmes et calculs à résoudre dans l'ordre. En cas de problème, n'hésitez pas à m'interroger pour ne pas rester bloqué. Ce TP sera évalué et vous donne l'occasion d'obtenir un bonus pour la note de l'examen final. Un rapport succinct (pas plus de 3 pages) vous est demandé (en format électronique uniquement). Ce rapport contiendra les réponses aux questions posées avec les justificatifs théoriques correspondant. Le code matlab est à ajouter en annexe de ce rapport. Envoyez le rapport à : `philippe.muellhaupt@epfl.ch`

2 Rappels théoriques

Commandabilité et linéarisation exacte

Pour un système linéaire décrit par l'ODE $\dot{x} = Ax + bu$, sa commandabilité est determinée par le rang de sa matrice de commandabilité

$$C = \begin{bmatrix} b & Ab & A^2b & \cdots & A^{n-1}b \end{bmatrix}$$

Si $\dim x = n$ et rang $C = n$, le système est commandable (voir Systèmes Multivariables). Comme vous l'avez vu dans le cours, vous savez qu'une notion similaire s'applique aux systèmes non-linéaires à une entrée décrits par une ODE dite affine en entrée, c'est à dire une dynamique du type $\dot{x} = f(x) + g(x)u$. Dans ce cas là, la matrice C est remplacée par une distribution C, que l'on peut décrire par une collection de champs de vecteur

$$C = \left\{ g,\, [f,g],\, [f,[f,g]],\, \cdots \operatorname{ad}_f^{n-1} g \right\}$$

comportant n termes. Lorsque C est de rang n dans un ensemble ouvert d'intéret, alors le système est dit accessible. Par continuité, il suffit d'évaluer C à $x = 0$ et d'évaluer le rang de la matrice obtenue pour vérifier l'accessibilité autour de l'origine.

Correspondance avec une chaîne d'intégrateurs

On se pose maintenant la question de savoir si l'on peut mettre le système donné en correspondance avec une simple chaîne d'intégrateurs à l'aide d'un changement de coordonnées et d'un bouclage d'état $v = \alpha(x)+\beta(x)u$ adéquats.

$$\begin{aligned} z &= \Phi(x) \\ v &= \Phi_u(x, u) \end{aligned}$$

Pour un système linéaire, la condition de rang sur la matrice C est suffisante. Pour un système du $\dot{x} = f(x) + g(x)u$, ce n'est pas toujours possible. Une condition supplémentaire dite d'"intégrabilité" est requise. On définit la distribution

$$\bar{C} = \left\{ g,\, [f, g],\, [f, [f, g]],\, \ldots,\, \mathrm{ad}_f^{n-2} g \right\}.$$

comportant $n-1$ termes. Comme C est de plein rang, n, $\bar{C}$ est de rang $n-1$. On peut donc trouver un annulateur de $\bar{C}$

$$\omega = \text{Null } \bar{C}$$

décrit par un vecteur ligne, c.-à-d. $\omega\bar{C} = 0$. (Formellement, ω est une 1-forme, vivant dans un espace vectoriel dont les éléments de base sont les formes linéaires $\{dx_1, dx_2, \ldots, dx_n\}$.) Maintenant, la condition d'intégrabilité mentionnée plus haut est donnée par l'intégrabilité de la forme ω.

Il y a deux moyens de vérifier l'intégrabilité de cette forme, l'une est de tester la condition de Frobenius $d\omega \wedge \omega = 0$. L'autre est indirecte, c'est l'involutivité de la distribution $\bar{C}$. En effet, on peut montrer que celle-ci est équivalente à la condition de Frobenius.

Si la chance nous sourit et que la condition d'intégrabilité est vérifiée, on sait que cela implique que le vecteur ligne ω est parallèle au gradient d'une fonction qu'il faut trouver. Autrement dit, il existe deux fonctions que l'on va écrire h, μ: $\mathbb{R}^n \to \mathbb{R}$ telles que

$$\omega_i(x_1, \ldots, x_n) = \mu(x_1, \ldots, x_n) \frac{\partial h(x_1, \ldots, x_n)}{\partial x_i}$$

Une fois le fonction $h(x)$ déterminée, on a que $z = h(x)$. La sortie plate du système est ainsi obtenue.

La procédure conduit aux identités suivantes (voir le cours pour les détails)

$$\begin{aligned} z_1 &= z = \Phi_1(x) = h(x) \\ z_2 &= \dot{z} = \Phi_2(x) = L_f h(x) \\ z_3 &= \ddot{z} = \Phi_3(x) = L_f^2 h(x) \\ &\vdots \\ z_n &= z^{(n-1)} = \Phi_n(x) = L_f^{n-1} h(x) \end{aligned} \tag{1}$$

et

$$v = L_f^n h(x) + L_g L_f^{n-1} h(x) u = \alpha(x) + \beta(x) u$$

Remarques : Les identités ci-dessus s'obtiennent en dérivant successivement $z = h(x)$ par rapport à t en remarquant que $\frac{dv(x)}{dt} = L_f v + u L_g v$. Dans les $(n-1)$ premières étapes, le terme en u s'annule par construction, puisque $\omega \cdot \bar{C} = 0$. Donc

$$\begin{aligned} u &= \frac{1}{L_g L_f^{n-1} h(x)} (v - L_f^n h(x)) \\ &= \frac{1}{\beta(x)} (v - \alpha(x)) \end{aligned} \tag{2}$$

ce qui avec (1) fournit la loi de bouclage linéarisante ! On peut donct dimensionner un régulateur linéaire sur les système $\frac{d^n z}{dt^n} = v$, c.-à-d. une loi de commande $v = -k^T z = -k_1 z_1 - k_2 z_2 \ldots - k_n z_n$ où k est un vecteur constant ce qui conduit à une loi de commande pour le système non-linéaire original en utilisant les lois de transformation (1) et (2).

TP

Le système étudié: le joint flexible

On se propose détudier la dynamique du joint flexible et de dimensionner un régulateur par linéarisation exacte. Les équations de la dynamique du système s'écrivent

$$I\ddot{q}_1 + MgL\sin(q_1) + k(q)(q_1 - q_2) = 0 \tag{3}$$

$$J\ddot{q}_2 - k(q)(q_1 - q_2) = u \tag{4}$$

La fonction de rappel du ressort $k(q)$ (que l'on modélise traditionnellement par une constante) est modélisée par la fonction non-linéaire

$$k(q) = K + (q_1 - q_2)^2$$

Question 1

Réécrire les équations de la dynamique sous la forme d'un système du premier ordre affine en entrée, c-à-d., sous la forme

$$\dot{x} = f(x) + g(x)u$$

Accessibilité et intégrabilité

Question 2

Construire la distribution

$$C = \{g,\, [f,g],\, [f,[f,g]],\, [f,[f,[f,g]]]\}$$

et

$$\bar{C} = \{g,\, [f,g]\, [f,[f,g]]\}$$

puis vérifier les conditions de rang et d'involutivité. (Vous pouvez travailler avec la toolbox symbolique de Matlab). Pour calculer les crochets de Lie, il est utile de créer une fonction. Utiliser la fonction Matlab `jacobian()`.

Sortie linérisante (plate)

Question 3

Construire une forme ω qui annule la distribution $\bar{C}$, c-à-d. chercher un vecteur ligne qui annule la matrice 4×3 formée par les éléments de $\bar{C}$. Utiliser la fonctiond de Matlab `null()`.

Question 4

Intégrer la forme $\omega(x)$. Pour ce faire, il faut trouver deux fonctions $\mu, h : \mathbb{R}^4 \to \mathbb{R}$ telles que

$$\omega_i = \mu \frac{\partial h}{\partial x_i} \qquad i = 1, 2, 3, 4.$$

En général, résoudre cette équation est non-trivial, mais vous verrez que dans notre cas, il suffit d'un tout petit peu d'imagination. $z = h(x)$ reprśente la sortie linéarisante.

Bouclage et stabilisation du système

Nous savons donc que notre système est équivalent à une chaîne d'intégrateurs dont l'équation dynamique est $z^{(n)} = v$ via les transformations d'état (1) et d'entrée (2). Une chaîne d'intégrateurs est un système linéaire commandable, donc facile à stabiliser.

Question 5

Ecrire la loi de commande du système non-linéaire $u = \kappa(x)$ correspondant à la loi $v = -kz$ du système linéaire équivalent à partir des équations de transformation (1) et (2). Utiliser le gain de régulation $k = (\ 1 \quad 4 \quad 6 \quad 4\ )$, ces valeurs sont obtenues avec la formule d'Ackermann, en plaçant tous les pôles à -1.

Question 6

Compléter le fichier Matlab et simuler le système non-linéaire bouclé avec différentes conditions initiales !

A. Quelques fonctions matlab utiles

- `lyap` : Résoud l'équation $A^T P + PA = -Q$ pour P. (Attention, il faut donner A^T et non A !)
- `help`: la fonction d'aide de Matlab
- `plot`: permet de créer une figure à partir d'un ou de deux vecteurs pour l'affichage.
- `length`: une fonction qui donne la longueur d'un vecteur
- `size`: une fonction qui donne la dimension d'une matrice ou d'un vecteur
- `ode45`: une fonction qui intègre une équation différentielle ordinaire.

- `N = null(A')'`: Retourne une matrice (ou un vecteur ligne) telle que `N*A=0`.
- `matlabfunction`: Transforme une expression symbolique en une fonction Matlab "normale" (beaucoup plus rapide que `eval()`)
- `jacobian`: Calcule les dérivées partielles par rapport à un vecteur de variables.
- `f = @(vars) expr`: Créer la fonction
- `f(vars) =` sans le fichier externe (très pratique).