

# TD - Commande robuste

Thao Dang

24 octobre 2011

Nous illustrons l'étude de la robustesse d'un contrôleur au travers de l'exemple du contrôleur d'orientation de robots LEGO. Le contrôleur de distance peut être analysé de la même manière.

Les équations différentielles qui décrivent la dynamique du robot sont :

$$\dot{x} = (v_g + v_d)\cos(\theta)/2 \quad (1)$$

$$\dot{y} = (v_g + v_d)\sin(\theta)/2 \quad (2)$$

$$\dot{\theta} = (v_d - v_g)/l \quad (3)$$

où  $x_c$  et  $y_c$  sont les coordonnées de la position courante le robot ;  $\theta$  est son orientation ;  $v_g$  et  $v_d$  sont les vitesses des roues à gauche et à droite ;  $l$  est la distance entre les deux roues.

Nous contrôlons l'**orientation** du robot par un contrôleur de type PI dont la fonction de transfert est

$$C_\theta(s) = \frac{ki_\theta}{s} + kp_\theta \quad (4)$$

où  $ki_\theta$  est le coefficient de l'action proportionnelle et  $kp_\theta$  est le coefficient de l'action intégrale.

Comme nous utilisons la différence entre les vitesses des roues pour contrôler l'orientation du robot, nous avons la variable de commande  $u_\theta = (v_d - v_g)$ . De l'équation (3) on obtient :

$$\dot{\theta} = u_\theta/l$$

ce qui donne en Laplace  $s\theta(s) = u_\theta(s)/l$  (en supposant que les conditions initiales sont 0). Alors, la fonction de transfert de  $\theta$  est

$$H_\theta(s) = \frac{\theta(s)}{u_\theta(s)} = \frac{1}{ls}.$$

Le schéma de commande est comme suit. La sortie  $u_\theta$  du contrôleur  $H_\theta(s)$  est connectée à l'entrée du bloc  $H_\theta(s)$  correspondant à la dynamique de l'orientation du robot. L'entrée du contrôleur  $C_\theta(s)$  est la différence entre le retour de la sortie  $\theta$  de  $H_\theta(s)$  et l'orientation désirée  $\theta^*$ . L'évolution de  $\theta$  contrôlée par la commande  $u_\theta$  peut donc être représenté par un système en boucle fermée dont la fonction de transfert est :

$$\begin{aligned} H(s) &= \frac{C_\theta(s)H_\theta(s)}{1 + C_\theta(s)H_\theta(s)} \\ &= \frac{\frac{kp_\theta}{l}s + \frac{ki_\theta}{l}}{s^2 + \frac{kp_\theta}{l}s + \frac{ki_\theta}{l}} \end{aligned}$$

### 0.0.1 Discrétisation de contrôleur

Pour implanter le contrôleur du modèle fourni, il faut le discrétiser. Si on choisit une période d'échantillonnage  $T$  et une approximation du premier ordre

$$s = \frac{z - 1}{zT}$$

où  $T$  est la période d'échantillonnage, on obtient la fonction de transfert en temps discret  $H(z)$ .

### 0.0.2 Système avec des paramètres incertains

**Exercice 1.** Nous allons étudier la robustesse du contrôleur d'orientation vis-à-vis aux paramètres tels que  $T$ ,  $ki_\theta$ ,  $kp_\theta$ ,  $l$ . Afin d'utiliser les critères pour les systèmes en temps continu, nous utilisons la transformation suivante qui ramène le cercle unitaire en  $z$  vers le demi-plan gauche en  $\eta$

$$z = \frac{1 - \eta}{1 + \eta}$$

Notons  $a = \frac{kp_\theta}{l}$  et  $b = \frac{ki_\theta}{l}$ , la fonction caractéristique en  $\eta$  est :

$$p(\eta) = (4 + 2aT + bT^2)\eta^2 + (-2aT - 2bT^2)\eta + bT^2$$

Nous pouvons maintenant traiter la variable  $\eta$  comme la variable  $s$  dans des fonctions de transfert en Laplace.

**Question 1.** En utilisant les fonctions matlab dans le fichier **hurwitz.m** fourni, on calcule la matrice de Hurwitz pour déterminer la stabilité du contrôleur pour des différentes valeurs de la période d'échantillonnage  $T$  et des coefficients du PI. **Réponse.** *En matlab, après donner des valeurs numériques (à tester) aux paramètres  $a$ ,  $b$  et  $T$ , on définit le polynôme  $p$  ci-dessus par un vecteur de ses coefficients :*

$$p = [(4 + 2 * a * T + b * T * T)(-2 * a * T - 2 * b * T * T)b * T * T]$$

Puis, on calcule sa matrice de Hurwitz :

$$H = \text{hurwitz}(p, \text{deg}P)$$

où  $\text{deg}P = 2$  est le degré de  $p$ .

Pour utiliser le critère de Hurwitz, on regarde les mineurs principaux de la matrice  $H$ . Pour assurer la stabilité de  $p$ , il faut que ces mineurs soient positifs.

**Exercice 2.** Quand il y a plusieurs paramètres, la méthode de tester des valeurs sur une grille dans l'espace de paramètres peut devenir très couteuse. On souhaite donc trouver une condition analytique pour la robustess du système en utilisant le critère Bialas. Pour appliquer ce critère supposons le contrôleur satisfait la condition suivante  $4 + 2aT + bT^2 = c^2$  où  $c$  est constant (puisque le degré de la partie nominale doit être plus grand que celui de la partie avec paramètre. Alors,

$$p(\eta) = c^2\eta^2 + 2\eta c + 1(-2aT - 2bT^2 - 2c)\eta + bT^2 - 1$$

On peut exprimer  $p$  comme suits.

$$p(\eta) = p_0(\eta) + qp_1(\eta)$$

où

$$p_0(\eta) = c^2\eta^2 + 2\eta c + bT^2 =$$

et

$$p_1(\eta) = \eta$$

et  $q = (-2aT - 2bT^2 - 2c)$ .

**Question 2** : En considérant  $q = (-2aT - 2bT^2 - 2c)$  comme paramètre, utiliser le **critère Bialas** pour trouver la condition de stabilité robuste.

**Réponse.** On va utiliser le script matlab suivant.

```
%% Programme matlab
l=0.2; %axe inter-roue
%% parametres du controleur d'orientation
alpha=1/l;
omega=1;
ksi=1;
ki_teta=omega*omega/alpha;
kp_teta=2*ksi*omega/alpha;
a = kp_teta/l;
b = ki_teta/l;

%% periode d'echantillonnage
T = 0.0000001;

%% on utilise les valeurs de a,b et T ci-dessus pour obtenir des valeurs
%% 'realistes' des constants
csq = 4 + 2*a*T + b*T*T;
c = sqrt(csq)
gamma = b*T*T

%% Decomposition de p
%p_0(s) = c^2 s^2 + 2 s s + gamma
%p_1(s) = s
%On voit que p0 a des coefficients positifs. Ensuite, on teste si p0
% est stable
p0 = [c*c 2*c gamma];
r = roots(p0)
```

```

% construire des matrices Hurwitz pour chaque polynome
H0= [ 2*c 0; c*c gamma];
H1= [ 1 0; 0 0];

%H0 = hurwitz(p0,2)
%H1 = hurwitz(p1,1)

H0inv = inv(H0);

M = -H0inv*H1;

lambda = eig(M)

```

A partir des valeurs propres

lambda

calculé par le programme, on déduit les conditions suivantes pour les paramètres  $a$ ,  $b$ ,  $T$ .

$$q_{min} \leq (-2aT - 2bT^2 - 2c) \leq q_{max}$$

où  $q_{max} = \lambda_{max}^+$  (la valeur propre positive la plus grande) et  $q_{min} = \lambda_{min}^-$  (la valeur propre négative la plus petite). Ensuite on doit tenir compte des conditions sur les valeurs choisies de  $c$  et  $\gamma$  dans le programme :

$$4 + 2aT + bT^2 = c^2$$

et

$$bT^2 = \gamma$$