ESISAR 5ème ANNEE D'ETUDES

Département : AUTOMATIQUE

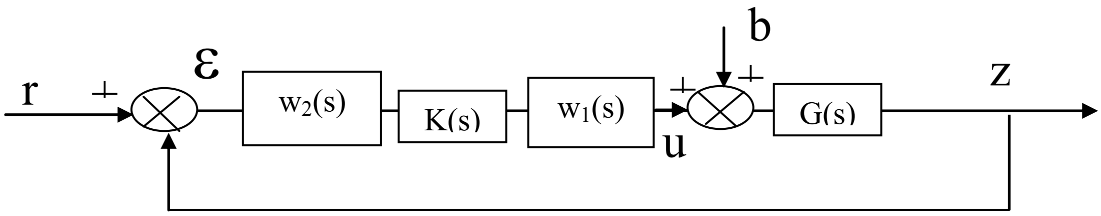
Auteur: Damien KOENIG

Module AC 560 : Synthèse H∞

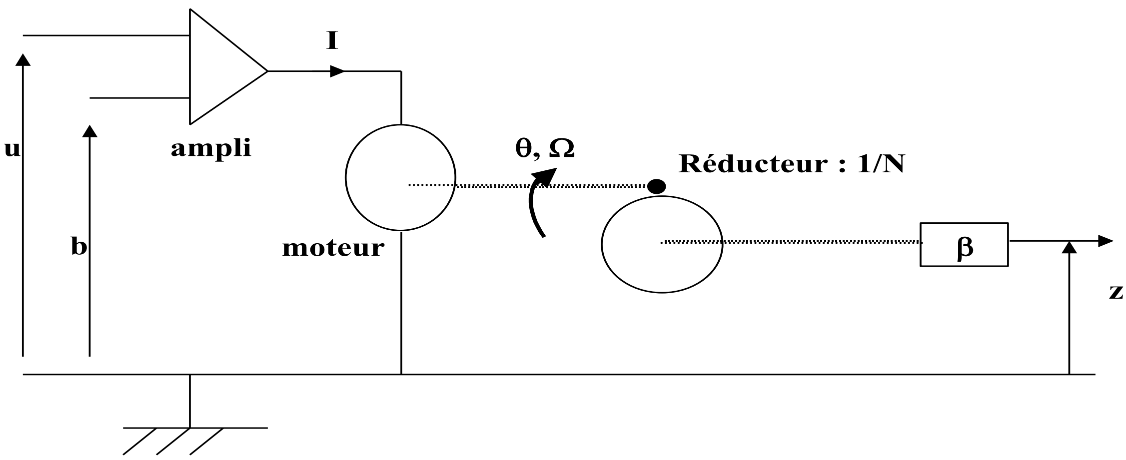
http://koenig-damien.jimdo.com

TP 2 : COMMANDE H∞ PAR MODELAGE DU TRANFERT DE BOUCLE (LOOP SHAPING) APPLICATION: MOTEUR A COURANT CONTINU

Introduction : Asservir en position un moteur à courant continu perturbé en entrée de commande par un offset et en sortie par des bruits. On appliquera pour cela la synthèse H∞ standard avec insertion de pré et post filtre dans la boucle de régulation. Le système étant SISO on rassemblera le pré et post filtre en un filtre W(s).



On considère le moteur à courant continu suivant :



où u est la tension de commande, b une perturbation constante (type tension offset) , z la mesure de position.

Les valeurs nominales des paramètres sont les suivantes

$$R = 5.8\Omega \quad J = 15.10^{-7}\,\mathrm{kg\,m^2} \quad A = 10 \quad L = 5.10^{-3}\,\mathrm{H} \quad a = 10^{-6}\,\mathrm{Nms} \quad N = 7$$

$$K_e = 0.024\,\mathrm{Vs} \quad K_c = 0.024\,\mathrm{NmA^{-1}} \quad \beta = 4\,\mathrm{Vrd^{-1}}$$

Objectifs de l'asservissement

- BP : ωc=100rd/s
- marges de module : > 0.4
- marges de stabilité : gain $\Delta G \cong 15$db, phase $\Delta\varphi \cong 50°$
- amplitude de la commande « raisonnable »
- erreur statique due à b faible < 1%

Modélisation : Les lois classiques de la mécanique et de l'électricité permettent d'établir le modèle suivant

$$\begin{cases} L\dfrac{dI}{dt} = -RI(t) - K_e\Omega(t) + A\big(u(t)+b\big) \quad ; J\dfrac{d\Omega}{dt} = K_c I(t) - a\Omega(t) \\ \Omega(t) = \dfrac{d\theta}{dt} \quad ; z(t) = \dfrac{\beta}{N}\theta(t) \end{cases}$$

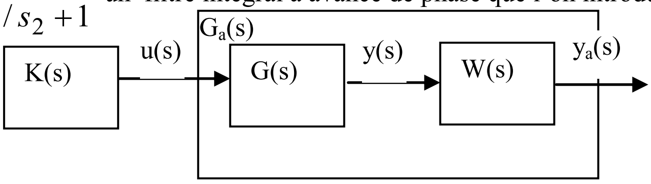
où la constante de temps L/R est $0.862.10^{-3}$ (p~1160).

1)Pourquoi peut-on négliger cette constante de temps (ou le mode) et considérer pour la synthèse du contrôleur le modèle du procédé suivant

$$G(s) = \frac{Z(s)}{U(s)} = \frac{240}{s(1+0.015s)}$$

2) Tracer l'évolution des valeurs singulière de G(s).

Soit $W(s) = \dfrac{k}{s}\dfrac{s/s_1+1}{s/s_2+1}$ un filtre integral à avance de phase que l'on introduit dans le tranfert de boucle.



3) Ajuster k, s_1 et s_2 tels que les valeurs singulières de $G_a(s) = W_2(s)G(s)W_1(s) = G(s)W(s)$ (voir rq 1.2)

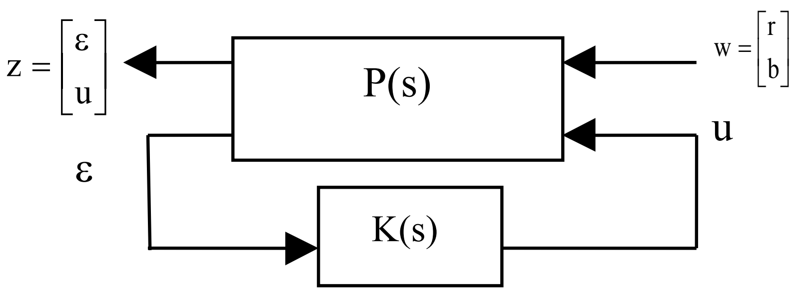
- présentent un gain élevé en BF (assurant ainsi le rejet des perturbations constantes et le suivi de consigne i.e. performance),
- une coupure de l'axe 0db à 100rd/s (BP souhaitée)
- et une atténuation en HF (robustesse des dynamiques négligées et bruits)

Tracer l'évolution des valeurs singulières de G(s)W(s) et G(s).

4) Résoudre le problème H∞ standard. On considèrera pour cela le procédé augmenté $G_a(s)$, la fonction de sensibilité $S = (1+KG_a)^{-1}$, la matrice de transfert

$$\begin{pmatrix} \varepsilon \\ u \end{pmatrix} \qquad \|M\|_\infty = \left\| \begin{pmatrix} S & SG_a \\ KS & KSG_a \end{pmatrix} \right\| < \gamma \qquad \begin{pmatrix} r \\ b \end{pmatrix}$$

et la réalisation P(s) issue du problème de synthèse H∞ standard, à savoir



Déduire alors à l'aide de la fonction matlab « hinfric », le régulateur K(s) et le majorant γ. Ajuster k, s1, s2 pour obtenir les marges souhaitées. Le correcteur final est bien entendu K(s)W(s).

5) Tracer l'évolution des valeurs singulières de la fonction de sensibilité : S(s)=1/(1+K(s)G(s)W(s))

6) Tracer l'évolution des valeurs singulières de la fonction complémentaire : T(s)=I-S(s)

7) Tracer le bode du correcteur K(s)W(s)

8) Tracer l'évolution des valeurs singulières : sensibilité* K(s)W(s)

9) Tracer l'évolution des valeurs singulières : sensibilité* G(s)

10) Commenter les différents tracés et conclure quant à l'intérêt de l'approche.

11) On pourra comparer les résultats avec la synthèse d'un contrôleur H2 avec pondérations fréquentielles définies par les filtres $R_c^{-1/2}(s)=\frac{1}{s/s_2+1}$ et $Q_c^{1/2}(s)=k\frac{s/s_1+1}{s}$ et associé à un observateur identité à pondérations unitaires, Qo=I, Ro=1.

Aide : les fonctions matlab utiles au problème posé sont :

tf, tfdata(G,'v'), tf2ss, ltisys, hinfric, slft, feedback, sigma, bode

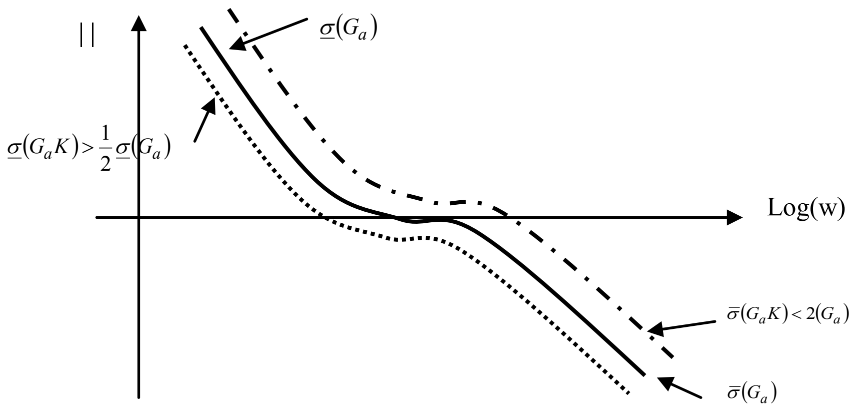
Remarques :

1. La première inégalité $\bar{\sigma}(S_a)<\gamma$ assure que la MM est supérieure à $\frac{1}{\gamma}$. En conséquence ajuster le filtre pour une marge de module > 1/2 implique un $\gamma<2$.
2. Pour des systèmes mono variable un seul filtre suffit, on ne parle pas de pré et post filtre.
3. En basse fréquence, généralement $|G_aK|>>1$, soit $|S_a|\cong\frac{1}{|G_aK|}<\gamma$, $|S_aG_a|\cong\frac{1}{|K|}<\gamma$, $|KS_a|\cong\frac{1}{|G_a|}<\gamma$ et $|KS_aG_a|\cong 1<\gamma$
   - => $\gamma>1$
   - => $\frac{1}{|K|}<\gamma \Leftrightarrow \frac{1}{\underline{\sigma}(K)}<\gamma \Leftrightarrow \frac{1}{\gamma}<\underline{\sigma}(K)$
   - => $\frac{1}{|G_aK|}<\gamma \Leftrightarrow \frac{1}{\underline{\sigma}(G_aK)}<\gamma \Leftrightarrow \frac{1}{\gamma}<\underline{\sigma}(G_aK)$ or $\frac{1}{\gamma}<\underline{\sigma}(K)$ et $\underline{\sigma}(G_aK)\geq\underline{\sigma}(G_a)\underline{\sigma}(K)$ soit $\underline{\sigma}(G_aK)\geq\underline{\sigma}(G_a)\underline{\sigma}(K)>\frac{1}{\gamma}\underline{\sigma}(G_a)$ =>

$$\underline{\sigma}(G_a K) > \frac{1}{\gamma}\underline{\sigma}(G_a)$$

4. En haute fréquence, généralement $|G_a K| << 1$, soit $|S_a| \cong 1 < \gamma$, $|S_a G_a| \cong |G_a| < \gamma$, $|KS_a| \cong |K| < \gamma$ et $|KS_a G_a| \cong |KG_a| < \gamma$
   - => $\overline{\sigma}(G_a K) < \gamma$ or $\overline{\sigma}(G_a K) < \overline{\sigma}(G_a)\overline{\sigma}(K)$ et $\overline{\sigma}(K) < \gamma$ soit $\overline{\sigma}(G_a K) \leq \overline{\sigma}(G_a)\overline{\sigma}(K) < \gamma\overline{\sigma}(G_a)$

Soit un tracé pour $\gamma = 2$ par exemple



Eléments de réponses sous matlab

```
G=tf([240], [0.015 1 0]); %procédé
...
Ga=G*GI*GAvph; %Etablir le système augmenté de l'intégrateur et de l'avance de phase
[numGa,denGa] = tfdata(Ga,'v') % 'v' signifie sys SISO
[Aa,Ba,Ca,Da]=tf2ss(numGa,denGa);
...
P=ltisys(Aa, [Bw Bu], [Cz; Cy], [Dzw Dzu; Dyw Dyu]); %réalise la réalisation P
[gopt,K] = hinfric(P,[1 1]); % determine K et gama
clsys=slft(P,K);  %  systeme BF
spol(clsys);      %  valeur propre de la BF
[numK,denK] = ltitf(K);
K_=tf(numK,denK);
Ka=K_*GI*GAvph; % correcteur augmenté du loop-shape
SysBF=feedback(Ka*G,1); % T=I-S
S=feedback(1,Ka*G);
Sens_bruitSortie_u=Ka*S;
Sens_b_eps=G*S;
%effectuer les tracés
...
```