

## TP2 : Régulation de la vitesse du moteur DC par les régulateurs PID

### 1. Objectifs

- Détermination de la fonction de transfert
- Savoir les procédures de calcul des gains du correcteur PI
- Maîtrise de l'outil Simulink du logiciel Matlab
- Confirmation des résultats simulés et théoriques

### 2 . Rappels théorique

#### 2.1. Régulateurs PID analogiques

Ce genre de correcteurs se trouve souvent dans l'industrie. La boucle fermée de régulation PID (P: Proportion, I : Intégration et D : Dérivation) suivante dont le design du correcteur PID est basé sur les spécifications désirées du système.

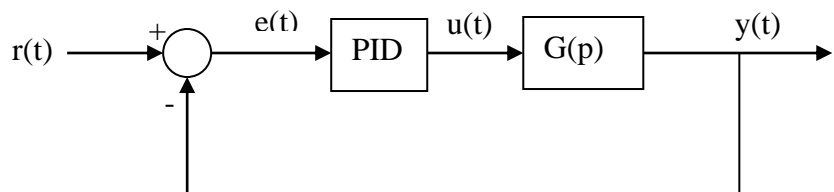
**P** → L'action **proportionnelle** corrige de manière instantanée, donc rapide, tout écart de la grandeur à régler, elle permet de vaincre les grandes inerties du système par l'ajout d'un gain. Le régulateur P est utilisé lorsque l'on désire régler un paramètre dont la précision n'est pas importante.

**I** → L'action **intégrale** complète l'action proportionnelle. Elle permet d'éliminer l'erreur résiduelle en régime permanent. L'action intégrale est utilisée lorsque l'on désire avoir une précision parfaite.

**D** → L'action dérivée, en compensant les inerties dues au temps mort, accélère la réponse du système. L'action D est utilisée dans l'industrie pour le réglage des variables lentes telles que la température.

$$P : u(t) = K_p \cdot e(t).$$

$$I : u(t) = K_I \int_0^t e(t) dt$$



$$D : u(t) = K_d \frac{de(t)}{dt}$$

La fonction de transfert du correcteur PID est comme suit :

$$C(p) = K_p + K_I \frac{1}{p} + K_d p$$

$K_p$ ,  $K_I$  et  $K_d$  sont les gains du correcteur PID

Il existe diverse méthodes de conception du régulateur PI. La méthode de lieu des pôles, la méthode de Bode et la méthode paramétrique A titre d'exemple, on se base sur la méthode de lieu des pôles.

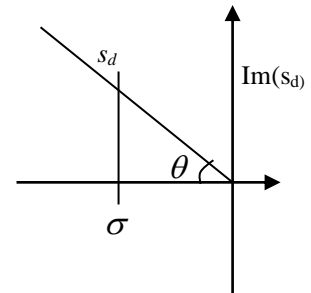
### **Méthode du lieu des pôles**

Etant donné les spécifications désirées du système compensé défini par le coefficient d'amortissement,  $\xi$  et le temps de réponse,  $t_r$ , la procédure de conception est donnée par ces étapes :

1. Obtenir la nouvelle forme de la fonction de transfert du système soit :

$$G(s) = Kd \frac{s + z}{s} \frac{\prod_{i=1}^m (s + z_i)}{s^l \prod_{j=1}^n (s + p_j)} \text{ avec } Kd = K_{pi} k$$

2. Déterminer l'intersection des deux demi droites montrées par la figure ci-dessous dont  $\xi = \cos \theta$  et  $\sigma = -3/t_r$ , cette intersection détermine les pôles dominants celui qui possède la partie imaginaire positive.



3. Utiliser ce pôle,  $s_d$  et l'équation d'angle pour déterminer la contribution en angle nécessaire du zéro (-z), donnée par :

$$\alpha = \pi - \sum_{i=1}^m \alpha_i + \sum_{i=1}^{n+l+1} \beta_i \quad \text{où} \quad \alpha_i = \text{tg}^{-1}(s_d + z_i) \quad \text{et} \quad \beta_i = \text{tg}^{-1}(s_d + p_i)$$

Le report de cet angle permet de déterminer la position du zéro, -z sur l'axe réel donnée par

$$z = |\sigma| + \frac{I_m(s_d)}{\text{tg}(\alpha)}$$

4. Calculer le gain,  $K_d$  qui procure les pôles désirés. Ce gain est donné par  $K_d = \frac{\prod_{i=1}^n |s_d + p_i|}{\prod_{j=1}^m |s_d + z_j|}$

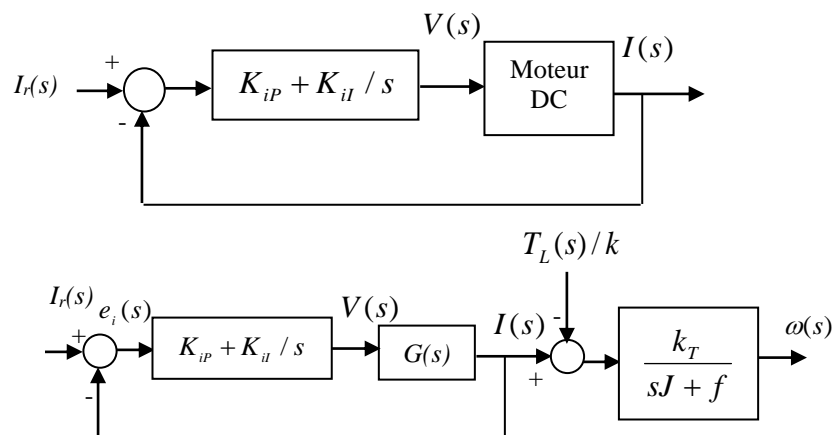
En déduire les paramètres du correcteur  $\begin{cases} K_{iP} = K_d / k \\ K_{il} = zK_{iP} \end{cases}$

### 3. Simulation à l'aide du logiciel Matlab

#### 3.1. Correction PI pour la boucle de courant

La correction PI est utilisée pour assurer une réponse indicielle apériodique ou oscillante, une erreur en régime permanent donnée, un temps de réponse donné et un système stable.

Le schéma block de la commande du courant en boucle fermée est montré par la figure ci-dessous.



**Figure. 1** Schéma bloc de commande de courant

Avec

$$G(s) = \frac{I(s)}{V(s)} = \frac{Js + f}{LJs^2 + (Lf + RJ)s + Rf + K_T^2}$$

$$= \frac{k(s + z_1)}{(s + s_1)(s + s_2)}$$

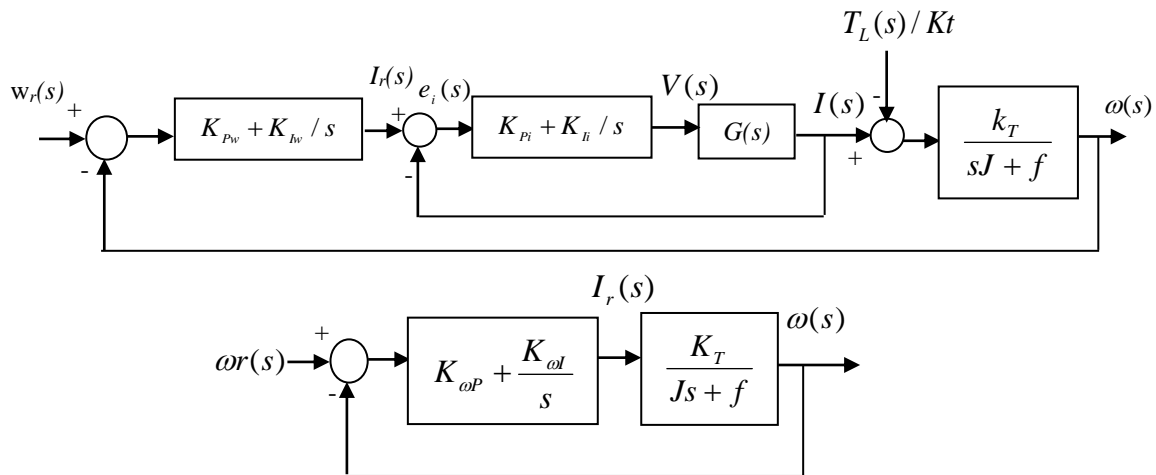
- En se basant sur le schéma ci-dessus, déterminé les pôles et les zéros de  $G(s) = I(s)/V(s)$
- Ecrire un programme sur Matlab pour déterminer les gains du correcteur PI, on utilisant la méthode du lieu des pôles avec les spécifications suivantes : erreur nulle en

régime permanent, un taux d'amortissement  $\xi = 0.3$  et un temps de réponse à 5% de 0.1 sec.

- En fonctionnement à vide et à partir de cette nouvelle réponse à un échelon, déterminer la valeur du dépassement et le temps de montée à 95%.
- Appliquant un couple résistant de 5 Nm et analyser la nouvelle réponse indicielle pour obtenir la vitesse en régime permanent, le dépassement et le temps de montée à 95%. Comparer vos résultats

### 3.2. Correction PI pour la boucle de vitesse

Le schéma block de la commande cascadée de vitesse en boucle fermée est montré par la figure ci-dessous.



**Figure. 2.** Schéma bloc de commande de vitesse

- En se basant sur le schéma ci-dessus, déterminé les pôles et les zéros de  $G(s) = \omega(s) / I(s)$
- déterminer les gains du correcteur PI, on utilisant la méthode du lieu des pôles avec les spécifications suivantes : erreur nulle en régime permanent, un taux d'amortissement  $\xi = 0.7$  et un temps de réponse à 5% de 0.4 sec.
- Pour un fonctionnement à vide avec une vitesse de référence égale à 1000 tr/min, analyser la réponse de la vitesse, le dépassement et le temps de montée à 95%.
- Comparer les résultats en variant le couple résistant jusqu'à 5 Nm à 1sec.
- Inverser le sens de rotation à  $t=1$ sec et tracer les allures de vitesse, du courant et de la tension.

- Changer la vitesse de référence et commenter les courbes simulées.
- En conclusion, donner l'intérêt de faire fonctionner le moteur en boucle fermée et le rôle du correcteur PI.

## Programme

```

clear all;
clc;
%Paramètres du moteur
R=1.2;L=0.18;Kt=0.284;
f=0.006;J=0.0024;

%Lieu des pôles.....

%courant
z1=roots([J f]);
roots([L*J (L*f+R*J) (R*f+Kt^2)]);
s1=-4.58-13.5*i;s2=-4.58+13.5*i;
xsi=0.3;
tr=0.1;
sigma=-3/tr;
theta=acos(xsi);
sd=sigma+tan(theta)*abs(sigma)*i;
alpha=pi-angle(sd+z1)+angle (sd)+angle(sd+s1)+angle(sd+s2)-2*pi;
asin(xsi)+pi/2;
z=abs(sigma)+(tan(theta)*abs(sigma))/tan(alpha);
K=abs(sd)*abs(sd+s1)*abs(sd+s2)/(abs(sd+z)*abs(sd+z1));
kp=K;
Ki=z*kp;

%Vitesse
xsi=0.7;
tr=0.4;
k=Kt/J;
s1=f/J;
sigma=-3/tr;
theta=acos(xsi);
sd=sigma+tan(theta)*abs(sigma)*i;
alpha=pi+angle(sd)+angle(sd+s1);
[alpha asin(xsi)+pi/2];
z=abs(sigma)+imag(sd)/tan(alpha);
Kd=abs(sd)*abs(sd+s1)/abs(sd+z);
kpw=Kd/k,
kiw=z*kpw,

```