

# REGULATEUR PID – BROÏDA

## I.1 Objectif :

L'objectif de ce TP est de réguler un système de premier ordre. Après l'identification du système en question, vous devez déterminer les caractéristiques d'un régulateur PID qui permettra d'atteindre les performances désirées du système. MATLAB sera utilisé durant ce TP comme outil d'analyse des relevés expérimentaux mais permettra également d'accélérer la phase de synthèse du correcteur.

## I.2 TRAVAIL EN BOUCLE OUVERTE

### I.2.1 Identification

Dans le cadre du contrôle d'un système industriel, une grande importance est accordée à l'identification du processus. En effet, pour procéder à la commande d'un système réel, il est nécessaire de disposer d'un modèle mathématique qui le caractérise avec une bonne approximation mais qui doit être suffisamment simple pour faciliter son étude.

Dans ce qui suit, vous allez établir un modèle mathématique du processus thermique étudié en utilisant la méthode d'identification empirique de Broïda. Elle se base sur l'allure de la réponse indicielle du système en BO. En effet, dans le cas où cette réponse indicielle ressemble à l'allure de la courbe décrite en figure 2, Broïda approche le comportement du système par une fonction de transfert de la forme suivante :

$$H(p) = \frac{S(p)}{U(p)} = \frac{K}{1 + \tau p} e^{-T_p p}$$

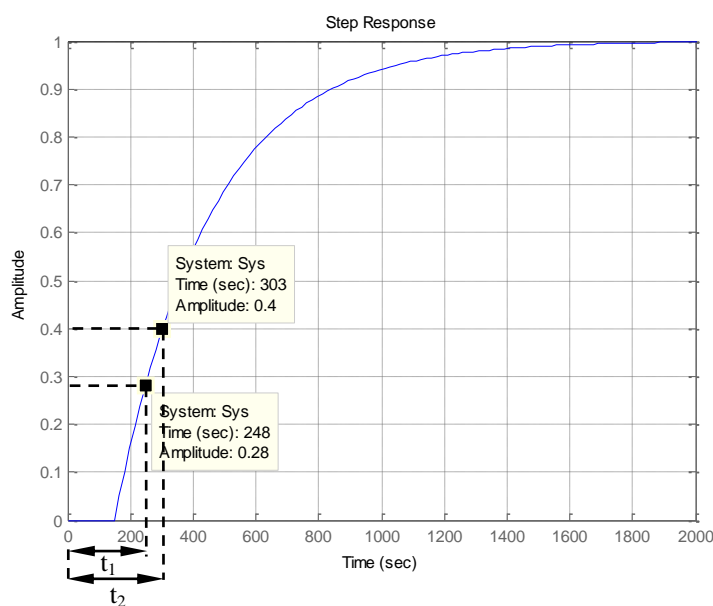


Figure 2. Exemple d'analyse de Broïda

- 1) Que représente  $e^{-Tp}$  ?
- 2) Que représente alors la fonction transfert  $H(p)$  vue dans sa globalité ?

Pour la détermination des paramètres  $\tau$  et  $T$  caractéristiques de  $H(p)$ , la méthode utilisée par Broïda consiste à s'appuyer sur deux points caractéristiques de la réponse du système. Les points habituellement choisis correspondent respectivement aux points d'ordonnées égales à 28% et à 40% de la valeur finale de la sortie. Ce qui se traduit par :

$$\frac{s(t)}{K.U_0} = 0.28 \Rightarrow \frac{t-T}{\tau} = 0.328$$

et

$$\frac{s(t)}{K.U_0} = 0.40 \Rightarrow \frac{t-T}{\tau} = 0.510$$

- 3) Démontrez les implications données ci-dessus.

En s'appuyant sur les valeurs numériques de  $t_1$  et de  $t_2$  obtenues expérimentalement (cf. figure. 2) on obtient :

$$t_1 - T = 0.328 \tau \quad \text{et} \quad t_2 - T = 0.510 \tau$$

La résolution de ces deux équations donne finalement :

$$\tau = 5.5 (t_2 - t_1) \quad \text{et} \quad T = 2.8 t_1 - 1.8 t_2$$

- 4) Effectuez les manipulations vous permettant de déterminer les paramètres  $t_1$  et  $t_2$  du système physique étudié. Déterminez les valeurs de  $K$ ,  $\tau$  et  $T$ .
- 5) Représentez dans une même figure la réponse indicielle du système réel en BO et celle du modèle identifié  $H(p)$ . Les fonctions MATLAB à utiliser sont :
  - « *tf* » pour créer la fonction de transfert,
  - *set(NomDeVotresysteme, 'OutputDelay', ValeurDuRetard)* pour approcher le retard pur « $T$ »,
  - «*step*» pour représenter la réponse indicielle du système.
- 6) Le modèle proposé est-il suffisamment fidèle ? Conclure.

### I.3 TRAVAIL EN BOUCLE FERMEE

Parmi la multitude de régulateurs utilisés en industrie, les régulateurs PID occupent incontestablement une place prédominante. En effet, les régulateurs PID présentent l'avantage d'être efficaces, mais aussi de pouvoir être utilisés même lorsque la fonction de transfert du système n'est pas exactement connue. Il existe alors des méthodes expérimentales pour le réglage des paramètres des PID. Une de ces méthodes est celle de Broïda, qui à partir des paramètres ( $K$ ,  $\tau$  et  $T$ ) du système identifié attribue les paramètres du PID ( $K_p$ ,  $T_i$  et  $T_d$ ) voulu selon le tableau ci-dessous.

Type de régulateur	P	PI	PID
K	$\frac{0.8\tau}{TK}$	$\frac{0.8\tau}{TK}$	$\frac{\tau + 0.4T}{1.2KT}$
T <sub>i</sub>	-	$K \frac{T}{0.8}$	$K \frac{T}{0.75}$
T <sub>d</sub>	-	-	$\frac{0.35}{K} \tau$

Tableau. 1.

La fonction de transfert d'un régulateur PID parallèle est la suivante :  $\frac{U(p)}{e(p)} = K_p + \frac{1}{T_i p} + T_d p$

La commande d'un système par un PID se résume par conséquent à : une commande proportionnelle à l'erreur (action P), on ajoute à cela une valeur moyenne des erreurs précédentes (action I), et on enlève une valeur proportionnelle à la variation de l'erreur (action D).

Dans chacun des essais en Boucle Fermée (BF) que vous allez réaliser dans ce qui suit veillez à relever à chaque fois les performances relatives à la régulation. En l'occurrence, l'erreur statique, le temps de réponse à 5%, le temps de montée, le dépassement ( $h_1/h_2$  [%]) (cf. figure. 3).

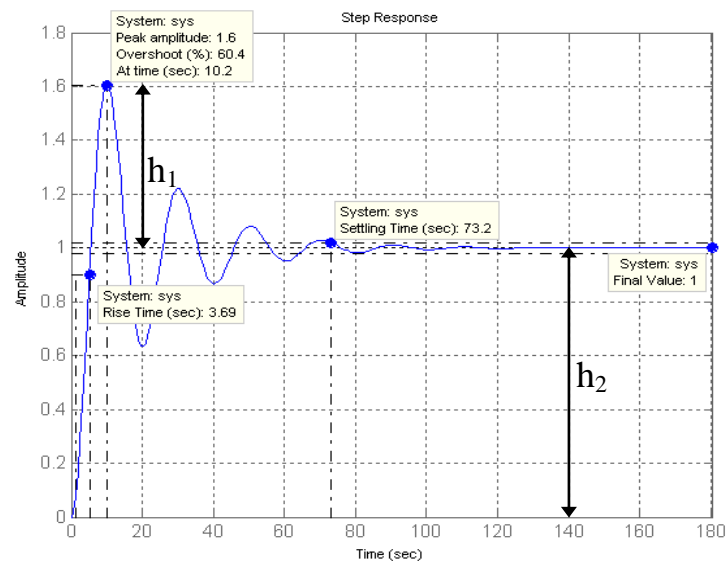


Figure. 3

- 1) Faites un essai en BF avec le régulateur proportionnel pur (P) déduit par la méthode Broïda. Observez et commentez les résultats (ne pas oublier de commenter l'évolution de la commande).
- 2) Mettez en place le régulateur PI et observez les résultats. L'erreur statique est-elle toujours présente ? Est-il possible d'améliorer la réponse du système ?

- 3) Observez les résultats avec la correction PID et commentez les performances. Pouvez-vous trouver un meilleur compromis de réglage des paramètres pour ce processus ? Commentez.

En plus des critères de performances temporelles relevés lors des expérimentations précédentes, il est important de juger des limites de stabilité du système régulé par une analyse fréquentielle. L'un des moyens pour quantifier cela est d'utiliser les diagrammes de Bode. Pour les systèmes lents comme les systèmes thermiques, l'acquisition expérimentale du diagramme de Bode n'est pas possible car les constantes de temps sont si longues que les fréquences d'étude doivent être basses et les temps d'acquisition seraient trop longs. Néanmoins, il est toujours possible de faire cette étude en simulation et d'extrapoler par la suite les résultats au système réel.

Vous allez étudier la marge de gain et de phase du système étudié en utilisant Simulink de MATLAB. Copiez pour cela le fichier Simulink (figure.4) présent dans le dossier «C:\TP\_Automatique» et mettez-le dans votre répertoire de travail. Remplacez  $\tau$  (noté Tau), K et T dans le modèle Simulink par leurs valeurs numériques obtenues précédemment.

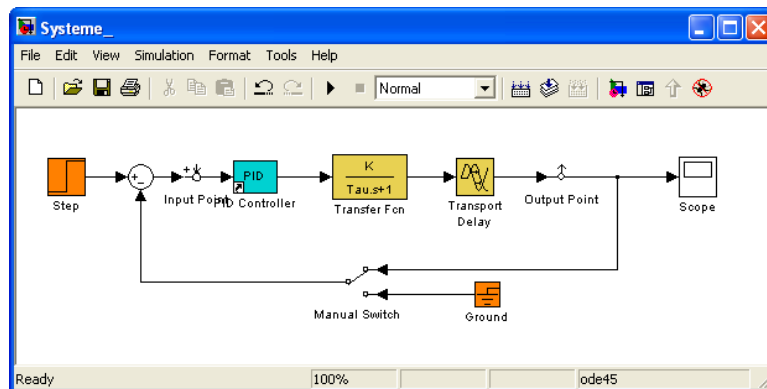


Figure. 4

- 4) Tout en gardant des performances temporelles satisfaisantes du système régulé, trouvez les paramètres du PID qui permettent d'obtenir une marge de gain ( $\Delta G$ ) et de phase ( $\Delta \rho$ ) du système au moins égales à : 10dB et à  $45^\circ$ .

**Remarque :**

Afin de faciliter votre étude vous allez utiliser LTI-Viewer qui permettra de visualiser d'une manière interactive différentes réponses du système à des signaux normalisés et surtout dans le cas de ce TP d'obtenir les diagrammes de Bode du système. Afin d'utiliser LTI-Viewer combiné avec votre système Simulink suivez les étapes suivantes :

- à partir de votre fenêtre Simulink : Menu Tools → Linear analysis,
- spécifiez les entrées-sorties du système que vous voulez étudier grâce aux nœuds fournis par LTI-Viewer (figure 5. (a)).
- à partir de la fenêtre LTI-Viewer : Menu Simulink → Get linearized model (figure. 5.(b)), qui va vous permettre de récupérer le modèle linéarisé d'entrée-sortie que vous venez d'exécuter sous Simulink,
- finalement, clic droit sur la fenêtre d'affichage de (LTI-Viewer) → Plot Types, et choisissez le type de réponse que vous voulez afficher (indicielle, impulsionnelle, Bode, etc.).

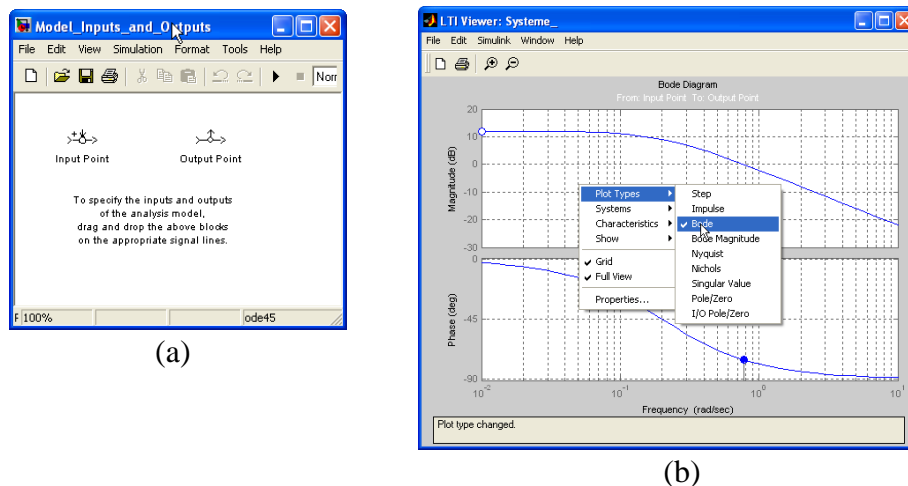


Figure. 5. Utilisation de LTI-Viewer avec Simulink

#### I.4 INJECTION DE PERTURBATION ET SATURATION DE LA COMMANDE :

- 5) Lancez un essai en BF avec les paramètres du PID obtenus avec la méthode de Broïda, lorsque le système est stabilisé, ouvrez la porte et laissez-la ouverte de façon à réaliser un échelon de perturbation. Que constatez-vous ? Conclure sur la réjection des perturbations du PID utilisé.
- 6) Simulez la présence de cette perturbation sur votre schéma Simulink. Que constatez-vous ? Conclure.
- 7) Ajoutez un bloc de saturation (Simlink Libraray → Discontinuités → Bloc Saturation) à la sortie du PID (figure. 6). A quelle limite devez-vous mettre vos seuils (haut et bas) pour vous rapprocher du modèle du système réel ? Appliquez-les, après quelques simulations en BF, que constatez-vous ? Conclure.

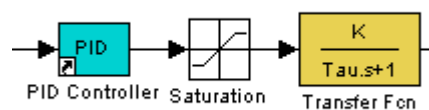


Figure .6

# MATLAB : ANNEXE

## QUELQUES FONCTIONS DE BASE POUR L'AUTOMATIQUE «CONTROL SYSTEM TOOLBOX»

### Représentation des systèmes :

MATLAB permet de représenter un système en utilisant les notions de fonction de transfert ou plus généralement de représentation d'état. Il est bien sûr possible de passer d'un mode de représentation à l'autre. Dans le cadre de cette annexe, c'est la représentation sous forme de fonction de transfert qui est retenue.

$sys = tf(num, den)$	Crée une fonction de transfert linéaire et continue « $sys$ » avec comme numérateur « $num$ » et comme dénominateur « $den$ ». $num$ et $den$ sont des vecteurs (polynômes) dont les coefficients sont rangés dans l'ordre des puissances décroissantes. $sys$ peut être par la suite utilisé comme un objet $tf$ ( <i>transfer function</i> ).
$sys.inputd = T$	Ajoute au système défini par $sys$ un retard pur de valeur $T$ secondes

### Interconnexion entre fonctions de transfert :

$sys = series(sys_1, sys_2)$ ou $sys = sys_1 * sys_2$	Crée le système $sys$ correspondant à la mise en série (cascade) de deux systèmes $sys_1$ et $sys_2$ .
$sys = parallel(sys_1, sys_2)$ ou $sys = sys_1 + sys_2$	Crée le système $sys$ correspondant à la mise en parallèle de deux systèmes $sys_1$ et $sys_2$ .
$sys_{BF} = feedback(sys, 1)$	Crée le système $sys_{BF}$ représentant le système $sys$ en boucle fermée (cas ici d'un retour unitaire).

### Information sur le système étudié :

$zpk(sys)$	Permet d'obtenir la forme factorisée de la fonction de transfert $sys$ . Ainsi il est possible de connaître la position des zéros et des pôles du système dans le plan complexe. Cette fonction permet également de connaître le gain statique de $sys$ .
$pole(sys)$	Donne les pôles du système
$zero(sys)$	Donne les zéros du système
$pzmap(sys)$	Affiche dans un graphique les pôles et les zéros du système dans le plan complexe.
$damp(sys)$	Donne les pôles ainsi que la pulsation propre et l'amortissement associés à chaque pôle du système $sys$ .

### Analyse temporelle du système étudié :

impulse ( <i>Sys</i> )	Trace la réponse impulsionnelle du système <i>Sys</i>
step ( <i>Sys</i> )	Trace la réponse indicielle du système <i>Sys</i>

**Analyse fréquentielle du système étudié :**

bode ( <i>Sys</i> )	Trace les diagrammes de Bode (du gain et de la phase) du système <i>Sys</i>
margin ( <i>Sys</i> )	Trace le diagramme de Bode et précise les marges de gain et de phase du système <i>Sys</i> .
nyquist ( <i>Sys</i> )	Trace le lieu de Nyquist du système <i>Sys</i>
nichols ( <i>Sys</i> )	Trace le lieu de Nichols (Black) du système <i>Sys</i>

**Outils de développements conviviaux et interactifs :**

ltiview ( <i>Sys</i> )	Permet une analyse autant temporelle que fréquentielle du système <i>Sys</i>
sisotool ( <i>Sys</i> )	Cette interface graphique permet la synthèse de correcteurs pour le cas des systèmes mono variables. Cela s'effectue grâce à une interactivité visuelle importante entre l'effet de la modification des paramètres du correcteur et l'évolution entre autres des lieux de Bode, de black-Nichols, etc.