

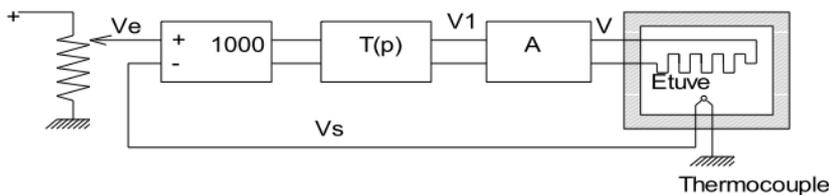
ASSERVISSEMENT DE TEMPERATURE D'UNE ETUVE POUR SECHER LES PATES

Exigence du CDC	Critères	Niveau
Acquisition d'une température fiable pour l'étuve	Précision statique	Ecart statique maximum : 1 % de la valeur de l'échelon
	Précision dynamique	Ecart de trainage maxi égal à 1°C
	Stabilité	Marge de gain 15 Db mini Marge de phase 45° mini



Photo de l'étuve installé chez mon voisin agriculteur qui possède un moulin et qui fait des pâtes sèches 😊

On désire étudier l'asservissement de température d'une étuve dont le schéma de principe est donné ci-dessous.



La température de consigne θ_e en degrés est affichée à l'aide d'un potentiomètre de fonction de transfert

$$V_e(p) / \theta_e(p) = 40 \mu V / ^\circ C$$

La température θ_s à l'intérieur de l'étuve est mesurée à l'aide d'un thermo-couple de fonction de transfert :

$$\frac{V_s(p)}{\theta_s(p)} = \frac{4 \cdot 10^{-5}}{1 + 2p}$$

Les deux signaux sont comparés et amplifiés à l'aide d'un préamplificateur différentiel de gain 1000.

Le signal délivré par le préamplificateur agit sur un bloc correcteur de transmittance $T(p)$ à déterminer ultérieurement. La tension V_1 en sortie de $T(p)$ commande l'amplificateur de puissance A de gain 10 qui agit sur la tension V aux bornes de la résistance chauffante de l'étuve.

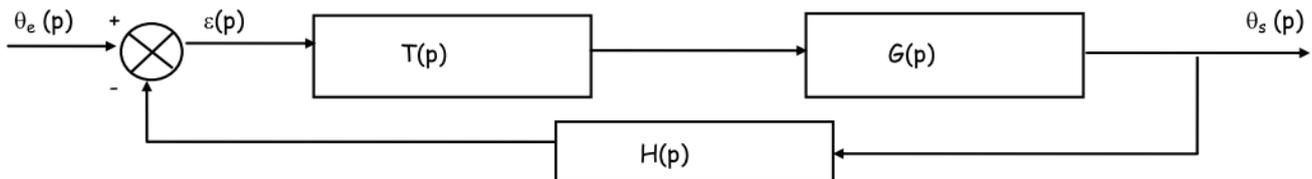
La fonction de transfert de l'étuve est donnée par la relation :

$$\frac{\theta_s(p)}{V(p)} = \frac{2}{1 + 125p} \text{ en } ^\circ\text{C/V}$$

1 MODELISATION

Modéliser sur Matlab l'asservissement sous forme de schéma-blocs.

Simplifier ce schéma pour lui donner la forme ci-dessous.



2 FONCTIONS DE TRANSFERT

Exprimer les fonctions de transfert

- de la chaîne directe
- de la boucle ouverte

3 ETUDE DE LA PRECISION.

$T(p)$ est un simple gain K . Déterminer en fonction de K l'expression de $\epsilon(p)/\theta_e(p)$ donnant l'écart en fonction de la température de consigne.

On applique à l'entrée un échelon de consigne θ_e °C.

En utilisant le théorème de la valeur finale des transformées de Laplace, déterminer, en fonction de K et de θ_e , la limite lorsque t tend vers l'infini (régime établi) de $\epsilon(t)$.

Calculer la valeur de K pour que cette limite soit égale à 1 % de la valeur de l'échelon θ_e °C.

$T(p)$ est maintenant de la forme : $T(p) = K / p$.

Déterminer la nouvelle expression de $\epsilon(p)$ en fonction de $\theta_e(p)$ et K .

Calculer la limite, lorsque t tend vers l'infini de $\epsilon(t)$, l'entrée étant toujours l'échelon θ_e °C.

Calculer la limite, lorsque t tend vers l'infini de $\epsilon(t)$, l'entrée $\theta_e(t)$ étant une rampe (échelon de vitesse) de pente 30°C / mn.

Calculer la nouvelle valeur de K pour que ce nouvel écart ne dépasse pas 1°C.

4 ETUDE DE LA STABILITE.

Dans la suite du problème, on prend $T(p) = 0,625 / p$

Tracer sur Matlab le lieu de transfert du système en boucle ouverte dans le plan de Bode et déterminer les valeurs des marges de phase et de gain. Vérifier le tracé asymptotique par calcul. Tracer ces marges sur les diagrammes.

Si ces marges en boucle ouverte sont insuffisantes, le système bouclé risque d'être instable.

Conclure sur la stabilité du système bouclé et relever la valeur de la pulsation critique ω_c pour laquelle le système bouclé risque de devenir instable.

5 FONCTION DE TRANSFERT DU CORRECTEUR QUI GARANTIT LA STABILITE SANS NUIRE A LA PRECISION.

On conserve la fonction $T(p)$ de la question 4 et on rajoute un bloc supplémentaire dont la fonction de transfert est

$$F(p) = \frac{1 + a\tau p}{1 + \tau p} \text{ avec } a > 1$$

La figure ci-contre donne une représentation dans le plan de Bode de $F(p)$ pour diverses valeurs de a en fonction de la pulsation réduite $u = \tau \cdot \Omega$

Afin de déterminer les caractéristiques du correcteur nous allons mettre en œuvre une méthode pratique qui évitera les itérations successives (cette méthodologie n'est pas à connaître)

On remarque que $F(p)$ laisse les phases pratiquement inchangées pour les valeurs de $u = \tau \cdot \omega$ inférieures à 0,01 ou supérieures à 10 mais qu'il produit une « avance de phase » pour les valeurs de u comprises entre 0,01 et 10.

On remarque aussi que l'avance de phase maximum est fonction de a .

Il suffit donc de superposer cette courbe de phase à celle du système décrit à la question 4 de façon que le maximum d'avance se situe au voisinage de la pulsation critique ω_c .

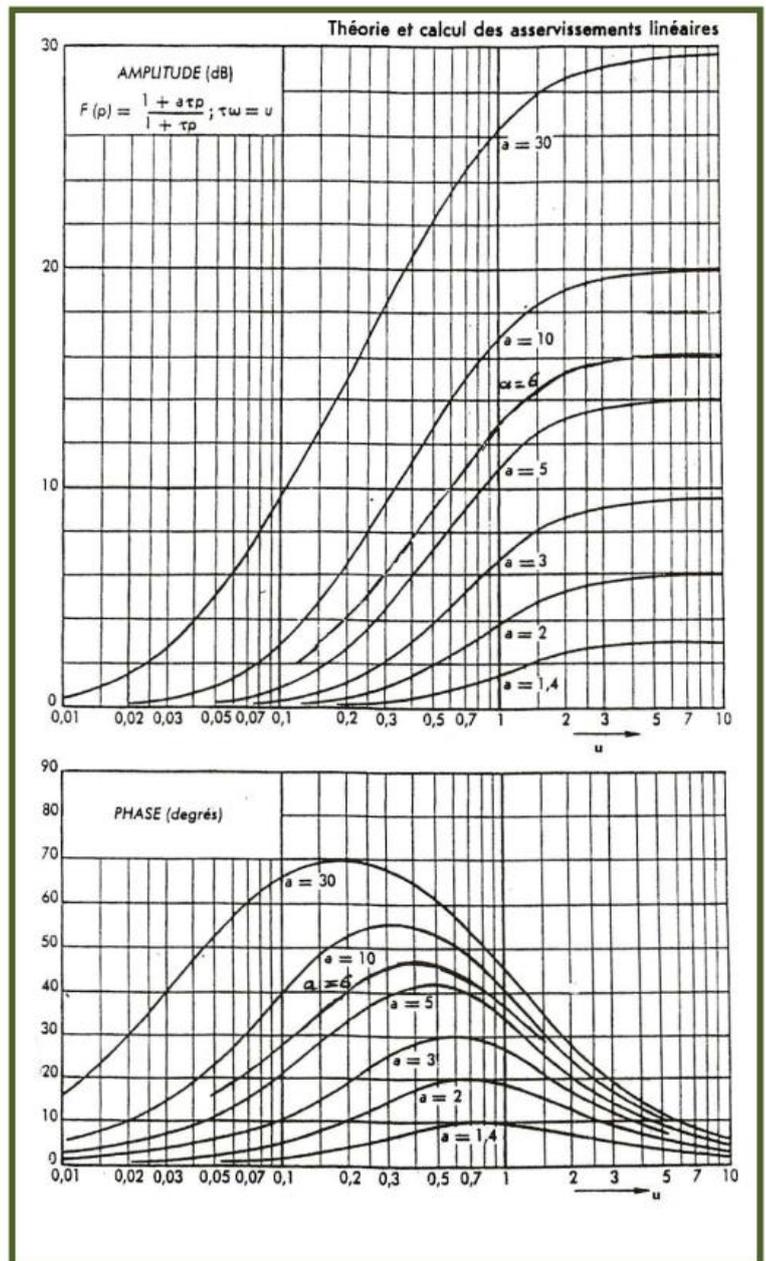
De plus, on remarque que l'amplitude de $F(p)$ n'est pas de 0 dB, ce qui aurait été le cas pour un correcteur parfait, mais qu'elle varie de 0 à un maximum qui dépend de a .

Il convient donc, en première approximation, de doubler environ la valeur de ω_c pour déterminer la pulsation donnant le maximum d'avance de phase et de majorer de 20% la valeur calculée de l'avance de phase maxi.

Exprimer la valeur de a qui confèrera à notre système une marge de phase de 45° et donner la valeur de $u = \tau \cdot \omega$ qui correspond au maximum d'avance.

Calculer alors la valeur de τ et donner la fonction de transfert $F(p)$ du correcteur dit à « avance de phase » ou correcteur « Proportionnel Dérivé ».

Tracer dans Matlab dans le même plan de Bode, d'une couleur différente, le lieu de transfert en boucle ouverte du système corrigé. **Donner** les nouvelles marges de phase et de gain.



6 CORRECTION NUMERIQUE

Parcourir *le document ressource* dans le dossier de prepabellevue pour mieux comprendre cette partie.

6.1 Implémentation d'un correcteur par les équations différentielles

Sur Python, nous allons implémenter les fonctions de transferts de la partie 2 et tester différents correcteurs.

L'implémentation d'un correcteur dans un micro processeur se fait grâce à un programme qui se décompose en 4 étapes répétées à chaque période d'échantillonnage :

- La lecture des données d'entrée (consigne et mesure pour un asservissement)
- Le calcul de l'écart, puis de la sortie, le plus souvent grâce à une équation aux différences ;

- L'écriture de la sortie ;
- Le vieillissement des données (les x_n deviennent les $x_n - 1$)

Lorsque la période d'échantillonnage est suffisamment faible devant la dynamique du système, on peut alors faire l'approximation suivante (différence finie centrée) :

$$\frac{dx}{dt} = \frac{x_n - x_{n-1}}{\Delta t} \text{ et } \frac{d^2x}{dt^2} = \frac{x_{n+1} + x_{n-1} - 2x_n}{\Delta t^2}$$

 **Méthode** Obtenir la relation de récurrence à partir d'une fonction de transfert

1. À partir de la fonction de transfert, retrouver l'équation dans le domaine de Laplace qui ne fasse intervenir que des multiplications par p ;
2. exprimer cette équation dans le domaine temporel ;
3. utiliser l'approximation sur la dérivée temporelle pour obtenir une équation aux différences ;
4. réorganiser les termes pour obtenir la nouvelle valeur de la sortie à l'échantillon n en fonction des valeurs de l'entrée à l'échantillon n et des valeurs de l'entrée et de la sortie aux échantillons précédents ($n - 1$, $n - 2$, etc.).

Ecrire l'équation différentielle associée à la FTBF de votre système et la récurrence qui en découle avec $T(p)=K$

Sur Python, créer une liste de consigne échelon de 16°C, puis appliquer votre récurrence pour calculer la sortie. Tracer les réponses en superposant la consigne et la sortie (FTBF)

Pour un correcteur proportionnel intégrale, l'équation s'écrit ainsi (avec $\epsilon(p)$ en entrée et $U(p)$ en sortie du correcteur)

$$C(p) = \frac{U(p)}{\epsilon(p)} = K_p \left(1 + \frac{1}{\tau_i p} \right) = K_p \frac{1 + \tau_i p}{\tau_i p}$$

1. $\tau_i p U(p) = K_p \epsilon(p) + K_p \tau_i p \epsilon(p)$;
2. $\tau_i \frac{du}{dt} = K_p \epsilon + K_p \tau_i \frac{d\epsilon}{dt}$;
3. $\frac{\tau_i}{T_e} (u_n - u_{n-1}) = K_p \epsilon_n + \frac{K_p \tau_i}{T_e} (\epsilon_n - \epsilon_{n-1})$;
4. $u_n = u_{n-1} + K_p \frac{T_e + \tau_i}{\tau_i} \epsilon_n - K_p \epsilon_{n-1}$;

On peut aussi le voir comme l'addition d'un correcteur proportionnel et d'un intégral.

$$i_n = i_{n-1} + \frac{K_p T_e}{\tau_i} \epsilon_n$$

$$u_n = K_p \epsilon_n + i_n$$

Pour le correcteur à avance de phase, l'équation devient :

$$C(p) = \frac{U(p)}{\epsilon(p)} = K_p \frac{1 + a\tau p}{1 + \tau p}$$

En repassant l'équation différentielle dans le domaine temporel, on obtient donc :

$$u + \tau \frac{du}{dt} = K_p \epsilon + K_p a \tau \frac{d\epsilon}{dt}$$

En passant à l'équation aux différences :

$$u_n + \frac{\tau}{T_e} (u_n - u_{n-1}) = K_p \epsilon_n + K_p a \frac{\tau}{T_e} (\epsilon_n - \epsilon_{n-1})$$

Équation que l'on peut factoriser ainsi ainsi :

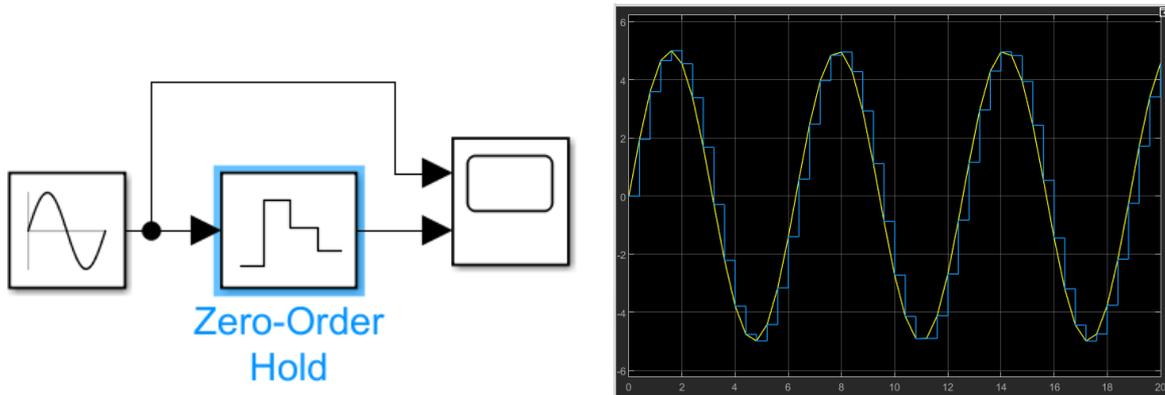
$$\left(1 + \frac{\tau}{T_e} \right) u_n = \frac{\tau}{T_e} u_{n-1} + K_p \left(1 + \frac{a\tau}{T_e} \right) \epsilon_n - K_p \frac{a\tau}{T_e} \epsilon_{n-1}$$

Le calcul de la sortie en fonction de la sortie précédente, de l'entrée et de l'entrée précédente sera donc :

$$u_n = \frac{\tau}{T_e + \tau} u_{n-1} + K_p \frac{T_e + a\tau}{T_e + \tau} \epsilon_n - K_p \frac{a\tau}{T_e + \tau} \epsilon_{n-1}$$

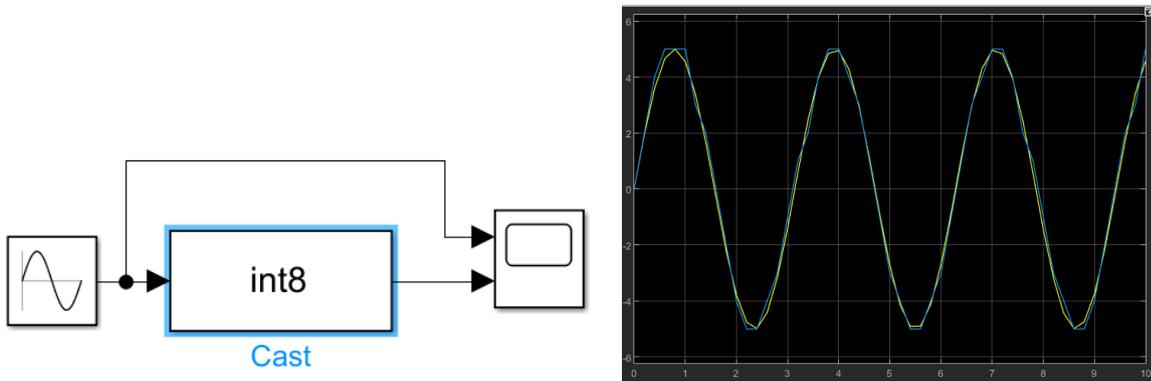
6.2 Echantillonnage et quantification

Sur Matlab, nous allons observer l'effet de l'échantillonnage et de la quantification sur la réponse et la stabilité. Tester le bloc échantillonneur-bloquer (*Sample-and-Hold*) comme sur le schéma suivant :



On parle également de bloqueur d'ordre 0 ou d'ordre 1 (*Zero-order hold ou first-order hold*) qui assure la continuité du signal par interpolation.

Tester le bloc de quantification (*quantization*) comme sur le schéma suivant :

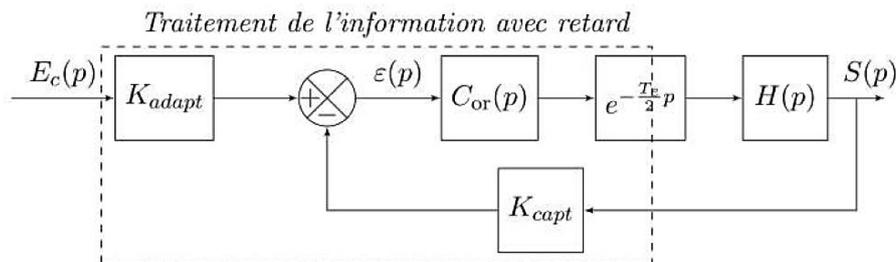


Ce bloc permet le réglage du quantum de la conversion, ainsi que la méthode de quantification.

Si on ne s'intéresse qu'aux cas extrêmes afin de visualiser ce qui se passe, un changement en entrée induit un changement en sortie :

- tout de suite après si le changement a eu lieu juste avant l'échantillonnage,
- au bout d'une période d'échantillonnage si le changement a eu lieu juste après le dernier instant d'échantillonnage.

Ainsi, en moyenne (ou en espérance), le signal d'entrée a un impact sur le signal de sortie au bout d'une demi-période d'échantillonnage. L'opération d'échantillonnage de l'information avec une période T_e peut donc être modélisée comme un retard de $\frac{T_e}{2}$. On peut donc continuer d'utiliser nos modèles linéaires et continus, avec les outils de Laplace, en ajoutant un retard pur de $\frac{T_e}{2}$ dans la chaîne de traitement de l'information d'un système asservi.



Le déphasage introduit par le retard pur tend à déstabiliser les systèmes en réduisant leur marge de phase aux hautes fréquences :

$$\arg(e^{-\frac{T_e}{2}j\omega}) = -\frac{T_e}{2}\omega \xrightarrow{\omega \rightarrow \infty} -\infty$$

L'intérêt de ce modèle est qu'il permet de garder l'hypothèse de continuité du système étudié, mais la fonction de transfert n'est plus une fonction rationnelle (ratio de polynômes en p). On pourra noter qu'il existe des approximations sous forme de fonctions rationnelles, parfois plus faciles à modéliser numériquement :

- $e^{-\frac{T_e}{2}p} \sim 1$ lorsque $\omega \leq \frac{2}{5T_e}$ (le retard pur est négligeable devant la dynamique du système) ;
- $e^{-\frac{T_e}{2}p} \sim \frac{1}{1 + \frac{T_e}{2}p}$ lorsque $\omega \leq \frac{1}{T_e}$ (approximation par un premier ordre) ;
- $e^{-\frac{T_e}{2}p} \sim \frac{1 - \frac{T_e}{4}p}{1 + \frac{T_e}{4}p}$ lorsque $\omega \leq \frac{4}{T_e}$ (approximation dite de Padé).

L'opération de quantification est non-linéaire. Selon ce que l'on souhaite retenir dans notre modélisation, on pourra retenir deux approches :

- approche globale : la plus commune, on néglige l'erreur de quantification, et la conversion analogique-numérique est traitée comme un gain, en volt/point (V/pt) ;
- approche locale : à l'intérieur d'un quantum, la valeur en sortie est constante. Cette approche est intéressante pour voir l'effet de la résolution sur le comportement du système : Quel est le changement minimum nécessaire pour induire une détection du système numérique ?