Ecole Nationale Supérieure de Physique de Strasbourg

Formation d'Ingénieurs en Partenariat

Automatique

TPs – Systèmes et asservissements à temps continu

Année 2008-2009

Pour être profitables, les travaux pratiques doivent faire l'objet d'une attention toute particulière. Il est fondamental d'avoir consacré du temps à la lecture du sujet ainsi qu'à sa compréhension avant la séance.

Contact des encadrants de TPs : ayadi@eavr.u-strasbg.fr, lott@eavr.u-strasbg.fr

Site du cours, archives, annales sur la page web de l'enseignant :
`http://eavr.u-strasbg.fr/~bernard`

Ce document évolue grâce à votre concours.
Pour l'améliorer, communiquez moi vos remarques ou corrections par mail :
bernard@eavr.u-strasbg.fr

Bernard Bayle

TP 1 – Initiation à Matlab

1 Principes de Matlab

Matlab est l'outil de référence pour la simulation numérique, notamment en ce qui concerne l'Automatique. Il offre des possibilités avancées que ce soit en matière d'identification ou de commande. Il permet, de manière plus générale, de résoudre une grande diversité de problèmes de simulation, dans des domaines aussi variés que le traitement du signal, les statistiques ou la vision, pour ne citer que quelques exemples. L'apprentissage de Matlab se fera en s'appuyant sur l'étude d'un moteur à courant continu.

1.1 Généralités

Avec Matlab les calculs sont numériques (une variable doit avoir une valeur) et basés sur la manipulation de scalaires, de vecteurs et de matrices.

Définir un scalaire Pour définir le réel $r = 2 * \pi$:

```
>>r=2*pi
```

Définir un vecteur Pour définir le vecteur $x = \begin{pmatrix} 1 \\ -1 \end{pmatrix} = (1 \quad -1)^T$:

`>>x=[1;-1]` ou `>>x=[1 -1]`

Définir une matrice Pour définir la matrice $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$:

`>>A=[1 2 3; 4 5 6]` ou `>>A=[1,2,3;4,5,6]`

Opérations sur les matrices (ainsi que sur les scalaires et les vecteurs, le cas échéant)
- addition : `A+B`
- soustraction : `A-B`
- multiplication : `A*B` et `B*A`
- inversion : `inv(A)`
- transposition : `transpose(A)` ou `A'`

Etude des matrices
- valeurs propres : `eig(A)`
- rang : `rank(A)`
- trace : `trace(A)`
- déterminant : `det(A)`

Quelques matrices utiles

– matrice identité de dimension n : `eye(n)`
– matrice de zéros de dimension $m \times n$: `zeros(m,n)`
– matrice de uns de dimension $m \times n$: `ones(m,n)`

De très nombreux algorithmes de calcul sont par ailleurs disponibles pour résoudre la plupart des problèmes courants (ou non ...) en mathématiques. Parmi ceux-ci, on peut citer les fonctions sur les polynômes (qui sont définis comme des vecteurs lignes composés des coefficients du polynôme entrés par ordre décroissant du degré). Par exemple, si on cherche les racines (complexes) de $x^2 - x + 1 = 0$:

```
>>roots([1 -1 1])
>>ans=
  0.5000 + 0.8660i
  0.5000 - 0.8660i
```

1.2 Aide en ligne

La bonne utilisation de l'aide en ligne est fondamentale pour travailler correctement avec Matlab.

Si l'on souhaite obtenir de l'aide sur certaines fonctions dont on connaît le nom, on utilise la fonction `help`. Par exemple (Matlab 7.1) :

```
>> help conv
CONV Convolution and polynomial multiplication.
C = CONV(A, B) convolves vectors A and B. The resulting
vector is length LENGTH(A)+LENGTH(B)-1.
If A and B are vectors of polynomial coefficients, convolving
them is equivalent to multiplying the two polynomials.

Class support for inputs A,B :
float : double, single

See also DECONV, CONV2, CONVN, FILTER and, in the Signal
Processing Toolbox, XCORR, CONVMTX.

Reference page in Help browser
doc conv
```

Si l'on cherche les noms des fonctions se rapportant à un sujet précis, on utilise la fonction `lookfor` (puis le mot en anglais). Par exemple :

```
>>lookfor polynom
>>POLYEIG Polynomial eigenvalue problem.
>>CONV Convolution and polynomial multiplication.
>>DECONV Deconvolution and polynomial division.
```

....

2 Utilisation de la Control Toolbox

2.1 Généralités

La boîte à outils dédiée à la commande (Control Toolbox) permet de disposer de nombreux outils d'analyse pour l'automatique.

Définition du système par sa fonction de transfert Soit le système décrit par :

$$G(s) = \frac{2s+1}{s^2+2s+1} = 2\frac{s+1/2}{(s+1)^2},$$

où s désigne la variable de Laplace. A l'aide de Matlab, on peut définir alternativement :

`>>F=tf([2 1],[1 2 1])` (numérateur et dénominateur de la fonction de transfert)
ou

`>>F=zpk([-1/2],[-1 -1],2)` (zéros, pôles et facteur de gain de la fonction de transfert)

Pour constituer un système à l'aide de différents sous-systèmes on peut effectuer différentes opérations. Soit `G1` et `G2` les représentations des deux systèmes. Les combinaisons de base sont :

`>>G1*G2`	ou	`>>series(G1,G2)`	`G1` en série avec `G2`
`>>G1+G2`	ou	`>>parallel(G1,G2)`	`G1` en parallèle avec `G2`
`>>feedback(G1,G2)`			`G1` bouclé par `G2`

On peut obtenir diverses informations sur le système défini par sa représentation G :

`>>pole(G)`	donne les pôles du système
`>>step(G)`	trace la réponse indicielle
`>>impulse(G)`	trace la réponse impulsionnelle
`>>bode(G)`	trace le diagramme de Bode
`>>nyquist(G)`	trace le diagramme de Nyquist
`>>nichols(G)`	trace le diagramme de Black-Nichols
`>>rlocus(G)`	trace le lieu d'Evans
`>>rlocfind(G)`	donne les valeurs des pôles et du gain correspondant sur le lieu d'Evans
`>>damp(G)`	donne les pôles, ainsi que la pulsation propre et l'amortissement associés à chaque pôle
`>>pzmap(G)`	place les pôles et les zéros dans le plan complexe

2.2 Prise en main de Matlab et de la Control Toolbox

On rappelle ici la modélisation du moteur à courant continu (MCC) dont l'étude va permettre d'illustrer les concepts fondamentaux de la Control Toolbox de Matlab. La fonction de transfert reliant la vitesse de rotation du rotor à la tension appliquée à l'induit s'écrit :

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{K}{(1+\tau_{el}s)(1+\tau_{em}s)},$$

avec :

$$\begin{aligned} K &= \frac{K_{em}}{Rf + K_{em}^2} \text{ gain statique du système,} \\ \tau_{em} &= \frac{RJ}{Rf + K_{em}^2} \text{ constante de temps électromécanique,} \\ \text{et } \tau_{el} &= \frac{L}{R} \text{ constante de temps électrique.} \end{aligned}$$

On rappelle que dans ce modèle R représente la résistance de l'induit du moteur, L son inductance ; f est le coefficient de frottement visqueux et J le moment d'inertie du rotor ; K_{em} est le rapport couple-courant (supposé égal au rapport force électromotrice-vitesse de rotation).

Pour la mise au point d'un programme ou des calculs très ponctuels, vous pouvez taper vos instructions sur la ligne de commande. Néanmoins :

A RETENIR – Dès que l'on a une séquence d'instructions a exécuter, on a tout intérêt à les regrouper sous forme d'un fichier script (fichier *.m).

Si un fichier a l'extension `.m` (par exemple `nomFichier.m`), alors il sera exécuté en tapant son nom (`>>nomFichier`) sur la ligne de commande.

1. Eteindre sa calculatrice et l'enfouir dans son sac pour le reste du TP (et des autres TPs d'ailleurs). En cas de non respect de cette consigne s'attendre à de graves représailles.
2. Créer un script qui comporte les différentes opérations détaillées ci-dessous Pour cela on peut utiliser l'éditeur de Matlab (`>>edit`). Des commentaires peuvent être introduits à l'aide du symbole `%`.
3. Définir tout d'abord les diverses constantes du problème (dans un script nommé calcul_constantes_mmc.m par exemple). Les valeurs numériques choisies correspondent à un MCC un Maxon F 2260, numéro 885 étudié en cours :

$$\begin{aligned} R &= 1,44\,\Omega \\ L &= 5,6\;10^{-4}\;H \\ J &= 1,29\;10^{-4}\;kg.m^2 \\ f &= 7,2\;10^{-5}\;m.N.s \\ K_{em} &= 0,10\;m.N.A^{-1} \end{aligned}$$

4. Calculer le gain K et les constantes de temps électrique τ_{el} et électromécanique τ_{em} du MCC. Définir alors sa fonction de transfert. Pour cela, on note `Num` et `Den` le numérateur et le dénominateur de la fonction de transfert.
5. Par la fonction appropriée, calculer les pôles de cette fonction de transfert. Vérifier que ces pôles valent $-\tau_{el}^{-1}$ et $-\tau_{em}^{-1}$.

6. Créer une figure (avec la fonction `figure`) et la diviser en deux sous-tracés (avec la fonction `subplot`). Dans le premier, tracer la réponse indicielle du MCC à un échelon unitaire de tension. A l'aide de la souris, observer les caractéristiques accessibles du tracé (clic droit puis relâcher pour les caractéristiques, pointer la courbe et clic gauche puis rester appuyé pour les valeurs). Dans la seconde sous-figure, tracer le diagramme de Bode du MCC. Analyser les différents tracés.
7. L'asservissement de vitesse du moteur à courant continu est défini à la figure 1. Il comporte un correcteur proportionnel de gain $K_p = 10$. La fonction de transfert du capteur de vitesse est assimilée à un gain pur noté K_ω. La sortie de ce capteur valant $10\ V$ pour une vitesse de rotation de $3000\ tours/min$ calculer K_ω en unités SI.

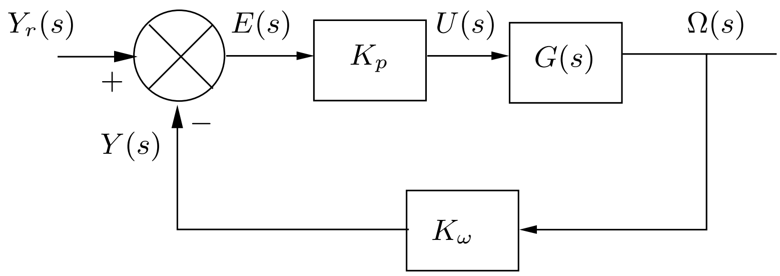


FIG. 1 – Schéma de l'asservissement de vitesse du MCC

8. Définir ensuite les fonctions de transfert du système en boucle ouverte et en boucle fermée en combinant les fonctions de transfert des différents blocs.
9. Sur une même figure, tracer les réponses indicielles du système en boucle ouverte et en boucle fermée.
10. Tracer les diagrammes de Bode, Black et Nyquist du système en boucle ouverte sur trois figures différentes (avec toujours $K_p = 10$). Identifier les marges de stabilité du système sur ces tracés.
11. En utilisant une boucle (`>>help for`), tracer sur un même graphe les réponses indicielles du système en boucle fermée pour les valeurs de K_p égales à $10,\ 100$ et 1000. Vérifier la cohérence de ces réponses avec les marges de stabilité relevées à la question précédente.

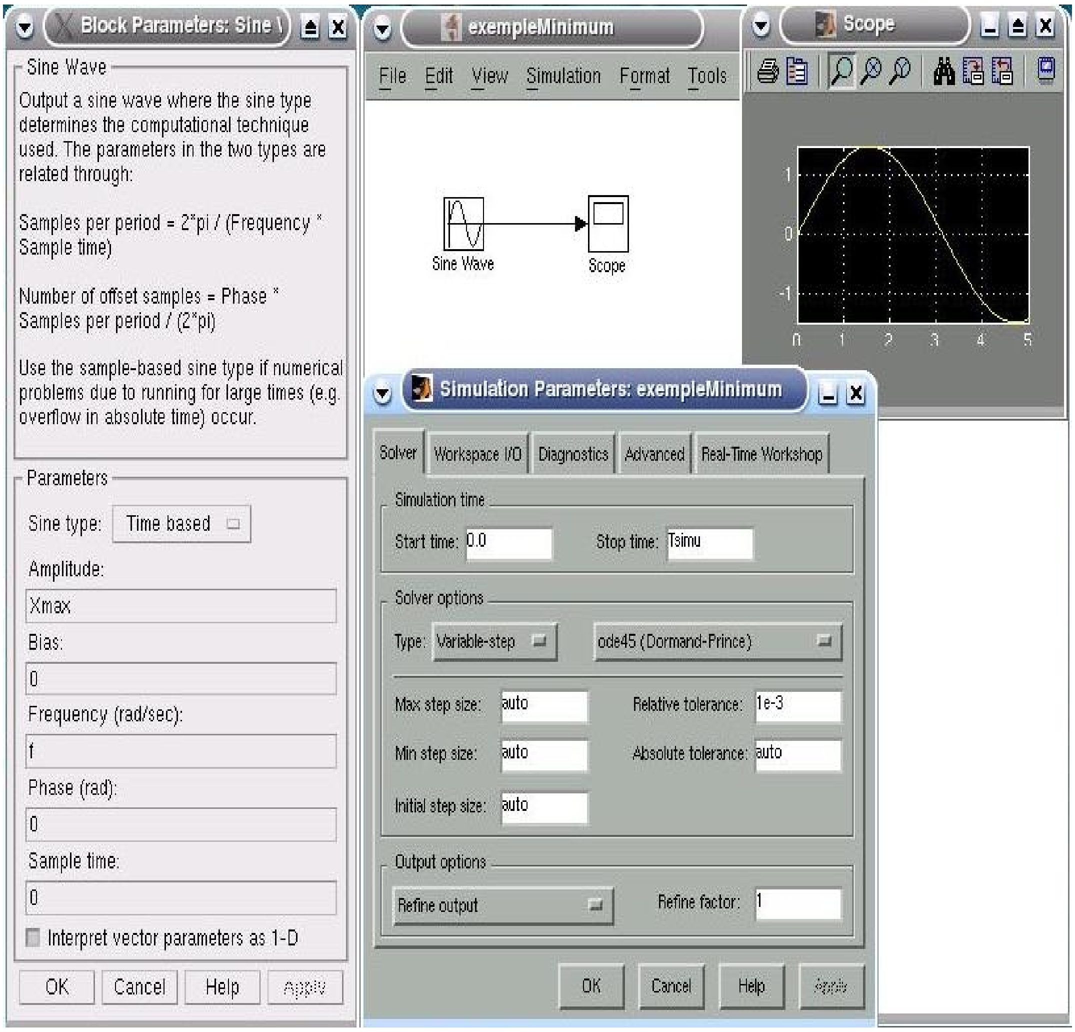
3 Utilisation de Simulink

3.1 Généralités

Simulink est une autre boîte à outils de Matlab qui permet de faire des simulations de systèmes définis à l'aide d'un outil graphique. On se propose ici d'utiliser Simulink pour définir l'asservissement en vitesse du moteur à courant continu. On pourra ainsi visualiser notamment les réponses du système à différents types d'entrées.

Pour lancer Simulink, on peut soit utiliser les menus disponibles, soit taper sur la ligne de commande `>>simulink`. Pour créer un nouveau modèle Simulink choisir `New` dans le menu `File`, puis `Model`. Une feuille de travail apparaît, sur laquelle on va pouvoir définir graphiquement notre système. Les différents outils disponibles seront trouvés dans les menus correspondants : sources, visualisation, automatique continue, automatique discrète, fonctions mathématiques, fonctions et tables, automatique non-linéaire, signaux et systèmes. De par sa nature graphique Simulink peut être aisément découvert intuitivement. Cet outil utilise la technique de *drag and drop* (sélectionner et faire glisser). Il est facile de positionner les éléments nécessaires dans la fenêtre du modèle. Ensuite, on relie ces éléments entre eux pour constituer le modèle. Chaque élément possède une description et éventuellement des paramètres qui peuvent être modifiés. Pour y accéder double-cliquer sur un élément.

Par exemple si on veut visualiser le signal d'un générateur sinusoïdal, on utilise la source correspondante (menu `Sources`) et un oscilloscope (menu `Sinks`). On connecte ensuite ces deux éléments en attrapant la sortie du générateur et amenant la souris enfoncée sur l'entrée de l'oscilloscope. La simulation est jouée en cliquant sur `Run`, dans le menu `Simulation`. La encore, on peut définir l'ensemble de la simulation à l'aide d'un script. En effet, Simulink partage les variables de l'espace de travail Matlab (variables globales). On peut ainsi définir le modèle Simulink à l'aide de variables dont les valeurs sont définies dans un script. On peut jouer la simulation depuis la ligne de commande (donc lancer cette simulation depuis un script). Ainsi sur l'exemple précédent, on obtient le modèle `exempleMinimum.mdl` et le script ci-après.



```
% Visualisation d'un signal sinusoïdal d'amplitude 1.5, de
% fréquence 1 Hz, sur un horizon de 5 s.
% le modèle simulé (voir ci-dessus) porte le nom exempleMinimum.mdl
Tsimu=5
Xmax=1.5
f=1
sim('exempleMinimum')
```

3.2 Prise en main de Matlab et de la Simulink

Selon la méthode décrite précédemment, on répondra aux questions suivantes, dont certaines reprennent largement l'étude effectuée pour la prise en main de la Control Toolbox. Néanmoins il est conseillé de créer un nouveau script.

1. A l'aide de Simulink créer le modèle de l'asservissement de vitesse vu dans la partie 2.2. On parcourra pour cela les menus de Simulink pour trouver les éléments nécessaires. En particulier le bloc fonction de transfert, nommé `Transfer Fcn` sera trouvé dans le menu `Continuous`. Les constantes, le numérateur et le dénominateur de la fonction de transfert seront définis dans le script qui pilotera les simulations, à l'image de l'exemple précédent.
2. Sur une même figure, tracer la consigne et la réponse indicielle du système en boucle ouverte. Le tracé sera fait sur un horizon de temps judicieusement choisi.

   Note : on peut envoyer à un oscilloscope autant de signaux que l'on veut. Par exemple, si l'on souhaite afficher deux signaux différents il faut utiliser un multiplexeur (`Mux` dans le menu `Signals and Systems`) pour les mettre sur une même ligne.
3. Définir le système en boucle fermée selon le schéma de la figure 1 avec $K_p = 100$. Sur une même figure, tracer la consigne et la réponse indicielle du système en boucle fermée.

TP 2 – Correction à avance de phase

1 Contexte de l'étude

On se propose de calculer un correcteur pour l'asservissement de position du moteur à courant continu étudié lors de l'initiation à Matlab.

La fonction de transfert reliant la position angulaire du rotor à la tension appliquée au moteur s'écrit :

$$\frac{\Theta(s)}{U(s)} = \frac{K_G}{s(1+\tau_e s)(1+\tau_{em} s)},$$

avec :

$$\begin{aligned} K_G &= \frac{K_{em}}{Rf + K_{em}^2} \text{ gain statique du système,} \\ \tau_{em} &= \frac{RJ}{Rf + K_{em}^2} \text{ constante de temps électromécanique,} \\ \text{et } \tau_e &= \frac{L}{R} \text{ constante de temps électrique.} \end{aligned}$$

On rappelle que dans ce modèle R est la résistance de l'induit du moteur, L son inductance ; f est le coefficient de frottement visqueux et J le moment d'inertie du rotor ; K_{em} est le rapport couple-courant (supposé égal au rapport force électromotrice-vitesse de rotation).

Le moteur dont l'asservissement de vitesse a déjà été étudié d'une part en cours et d'autre part lors de l'initiation à Matlab, est maintenant asservi en position selon le schéma de la figure 2. Dans ce schéma $y(t)$ représente la tension image de la position angulaire mesurée par un capteur potentiométrique. La fonction de transfert correspondante, moyennant un réglage correct de l'offset du capteur, est assimilée à un gain constant $K_\theta = 5,56\ 10^{-2}$ V . deg^{-1}, ce qui correspond à une amplitude de sortie de 20 V ($\pm$ 10 V) pour un tour complet du rotor du moteur ($\theta(t)$ étant exprimé en degrés).

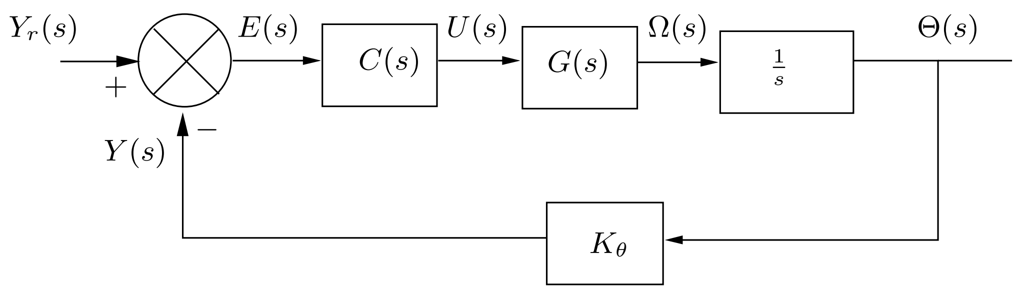


FIG. 2 – Asservissement de position d'un moteur à courant continu

2 Synthèse d'un correcteur

2.1 Correction proportionnelle et régime critique

Avant d'effectuer la synthèse d'un correcteur à avance de phase au paragraphe 2.2, on se propose tout d'abord de résoudre les questions suivantes, relatives à la correction proportionnelle du système (éventuellement traitées en cours si le temps l'a permis) :

1. Que vaut la valeur k_{pc} du gain d'un correcteur proportionnel assurant, en boucle fermée, la valeur limite de stabilité. Vérifier ce résultat avec Matlab, en traçant le diagramme de Bode du système en boucle ouverte et la réponse indicielle du système en boucle fermée.
2. Pour un correcteur proportionnel de gain k_{pc}, évaluer avec Matlab la pulsation de coupure du système.
3. Calculer le gain du correcteur proportionnel permettant d'obtenir une marge de phase de 45 deg. Vérifier ce résultat avec Matlab (fonction `margin`).
4. Représenter le lieu des racines (ou lieu d'Evans, obtenu à l'aide de la fonction `rlocus`) du système. Retrouver le résultat précédent, sachant qu'un amortissement légèrement supérieur de 0,4 conduit à une marge de phase de 45 deg.

2.2 Correction à avance de phase

Pour obtenir de meilleures performances dynamiques, on se propose d'asservir le moteur en position avec un correcteur à avance de phase (correcteur PD approché) sous la forme :

$$C(s) = k_p \, \frac{1 + aT_d s}{1 + T_d s},$$

où $a > 1$.

1. Représenter le diagramme de Bode asymptotique de ce correcteur puis tracer le diagramme réel avec Matlab (valeurs de a et T_d quelconques). Tracer par ailleurs le diagramme de Bode du système en boucle ouverte. Analyser l'influence d'un tel correcteur sur la marge de phase et sur la pulsation de coupure du système en boucle ouverte. Superposer au diagramme de Bode du système tracé précédemment le diagramme du système corrigé par un correcteur à avance de phase placé de sorte que l'on ait $\tau_e < T_d < \tau_{em}$ et $a > 10$.
2. Transposer les résultats du cours sur le correcteur à avance de phase à la forme de correcteur proposée ici (qui n'est pas la même au cas où vous le l'auriez pas remarqué) : donner la phase maximale apportée et la pulsation correspondante, en fonction de a et T_d. Vérifier votre calcul avec Matlab en traçant le diagramme de Bode du correcteur ainsi calculé.
3. Connaissant le gain nécessaire k_{pc} pour avoir le régime critique avec un correcteur proportionnel, calculer les paramètres a et T_d du correcteur à avance de phase de façon à ce que le système ait une marge de phase de 45 deg en boucle ouverte. Vérifier la marge de phase obtenue avec Matlab. Conclusions.

4. Effectuer la synthèse de ce même correcteur avec la méthode du lieu des racines. Pour cela, on se fixera un cahier des charges correspondant au critère de la question précédente, mais défini en termes de dépassement et de temps de réponse. Ensuite, on utilisera l'outil `rltool` (taper `rltool`, lire l'aide et se laisser guider) pour obtenir le correcteur répondant au cahier des charges voulu.
5. Calculer le gain ainsi que les paramétres a et T_d du correcteur à avance de phase de façon a avoir une marge de phase de 45 deg pour une pulsation de coupure du système en boucle ouverte de 300 rad/s. Vérifier le résultat obtenu avec Matlab.

TP 3 – Autoréglage de correcteurs PID

Cette étude concerne la réalisation de l'asservissement de température d'un système de chauffage électrique ventilé caractérisé par :
- une constante de temps lente ;
- un retard dû au transport de l'air.

Toutes les questions des parties *Identification* et *Calcul d'un correcteur PID* doivent être préparées en utilisant les figures réponses jointes en annexe. En séance les résultats correspondants seront vérifiés à l'aide de Matlab, avant d'aborder la partie *Autoréglage*.

1 Identification

1. Retrouver en la démontrant la transformée de Laplace du signal $g_1(t-t_r)$ avec t_r une constante réelle positive. On note $G_1(s)$ la transformée de Laplace du signal $g_1(t)$.
2. Donner la fonction de transfert $G_2(s)$ d'un système du second ordre possédant un pôle réel double. On introduira pour cela la constante de temps τ associée au pôle et le gain statique du système, noté K.
3. Que vaut le coefficient d'amortissement ξ d'un système du second ordre possédant un pôle double réel ? Justifier la réponse sans pour autant faire de calcul.
4. En utilisant la figure 3 jointe en annexe, donner le temps de réponse $t_{5\%}$ d'un tel système en fonction de τ. On utilisera pour cela la forme canonique de la fonction de transfert, rappelée dans l'annexe A.
5. On assimile le système de chauffage électrique ventilé considéré à un second ordre avec retard possédant un pôle double réel. D'après les questions 1 et 2, donner l'expression de sa fonction de transfert $G(s)$ en conservant les notations précédentes.
6. Tracer l'allure de la réponse indicielle correspondant à la fonction de transfert précédente en fonction de t_r, τ et K.
7. La réponse du système à un échelon d'amplitude $10V$ est donnée à la figure 5. Comparer cette réponse au tracé précédemment effectué et, d'après les questions précédentes, identifier les paramètres t_r, τ et K en utilisant la figure réponse 5.
8. On se propose d'identifier les paramètres précédents de manière plus précise en utilisant la méthode de Strejc. Son principe est expliqué dans l'annexe B. Appliquer la méthode de Strejc, en vous appuyant sur la figure réponse 6.

2 Calcul d'un correcteur PID

On se propose maintenant de réaliser un correcteur pour asservir le système de chauffage. Pour cela, on considérera que le système a pour fonction de transfert :

$$G(s) = \frac{2,4\,e^{-0.5s}}{(1+2s)^2}.$$

1. Le système étant relativement lent, on propose tout d'abord de choisir un correcteur permettant de compenser le pôle double du système. En déduire le choix du numérateur de la fonction de transfert du correcteur.
2. On souhaite ensuite obtenir une erreur statique nulle. Combien de pôles faut-il placer pour cela et où ? A quoi cela correspond-il ?
3. Pour que le correcteur soit causal, il faut compléter la synthèse par le placement d'un pôle. On obtiendra ainsi un correcteur à deux pôles et deux zéros. On propose de placer le pôle restant de façon à ce qu'il soit 20 fois plus rapide que le pôle double du système non corrigé. Que vaut alors ce pôle du correcteur et quelle est la constante de temps associée ?
4. On propose de faire le réglage du gain du correcteur selon une méthode fréquentielle. Tracer tout d'abord le diagramme de Bode en boucle ouverte du système corrigé :
$$CG(s) = \frac{2,4}{s(0,1s+1)}$$
sur la figure réponse 7.
5. On s'intéresse maintenant à l'influence du retard. Quel est le déphasage introduit par le terme de retard, à une pulsation ω donnée ?
6. Evaluer ce déphasage pour différentes valeurs de ω judicieusement choisies dans la plage de variation des pulsations de la figure 7. En déduire le diagramme de Bode complet de la fonction de transfert corrigée en boucle ouverte :
$$CG(s) = \frac{2,4\, e^{-0.5s}}{s(0,1s+1)}$$
dans la limite de la place disponible à la figure réponse 7.
7. Déterminer la marge de phase du système, et en déduire le gain proportionnel K_p permettant d'obtenir une marge de phase de 60 deg.

3 Autoréglage

Le but de cette partie est de réaliser une simulation à l'aide de Simulink permettant d'obtenir un autoréglage par le test du relai.

1. Réaliser un modèle Simulink pour mettre en œuvre l'auto-réglage par la méthode du relai. On choisira un relai avec hystérésis.
2. Obtenir des oscillations entretenues et mesurer les différentes caractéristiques nécessaires à l'autoréglage.
3. Implanter l'algorithme d'autoréglage dans un script Matlab, pour calculer un correcteur PID permettant d'obtenir une marge de phase de 60 deg à partir des mesures précédentes. On ne demande pas que ces mesures soient automatiques : les valeurs obtenues à la question précédente seront reportées dans ce script comme des paramètres.
4. Vérifier avec Matlab que la marge de phase obtenue est conforme au résultat attendu.
5. Observer le comportement du système corrigé et commenter le résultat obtenu.

A Temps de réponse d'un système du second ordre

L'abaque suivante donne le temps de réponse normalisé $\omega_n t_{5\%}$ d'un système du second ordre de fonction de transfert :

$$\frac{Y(s)}{U(s)} = \frac{K\omega_n^2}{\omega_n^2 + 2\xi\omega_n s + s^2}$$

en fonction de l'amortissement ξ.

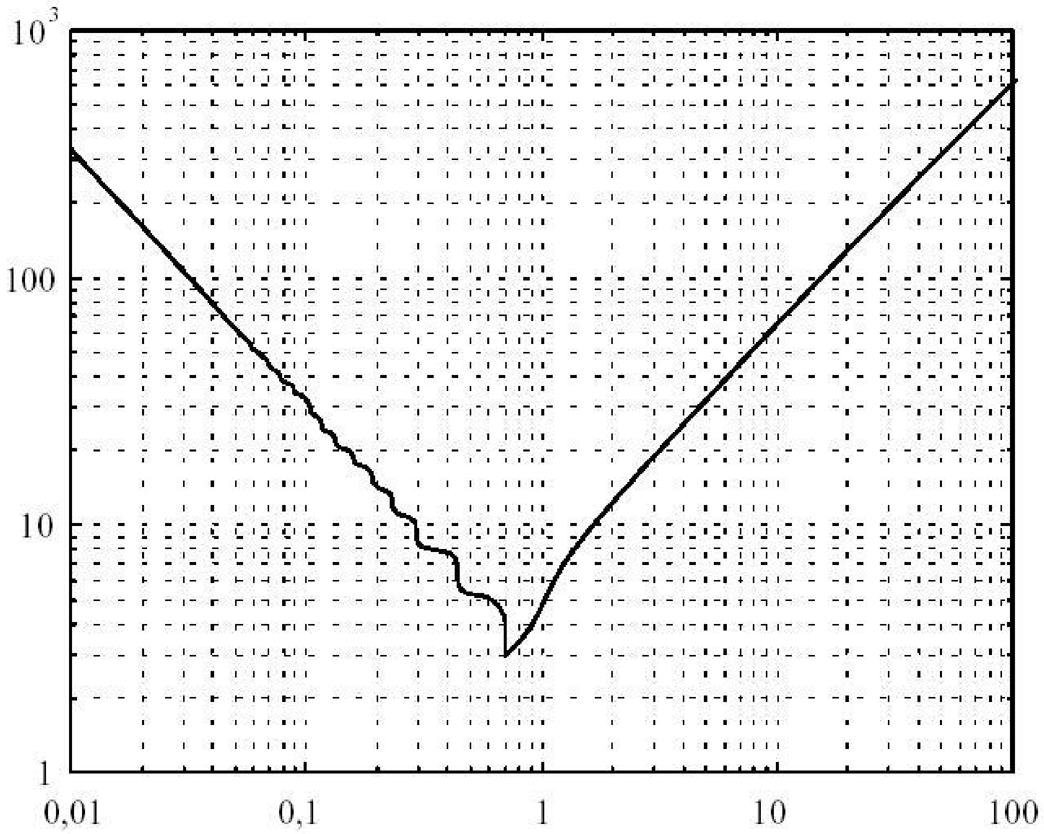


FIG. 3 – Temps de réponse normalisé $\omega_n t_{5\%}$ d'un système du second ordre en fonction de son coefficient d'amortissement ξ

B Méthode de Strejc

La méthode de Strejc est une méthode d'identification classique qui permet de déterminer les paramètres d'un système dont le modèle peut s'écrire sous la forme de la fonction de transfert :

$$G(s) = \frac{Ke^{-t_r s}}{(1+\tau s)^n}. \quad (1)$$

A partir de la courbe de la réponse indicielle (voir figure 4) on trace la tangente à la réponse

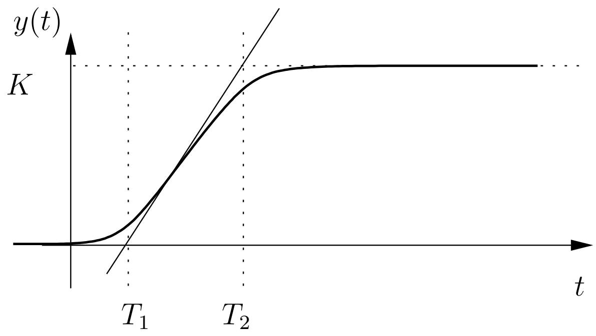


FIG. 4 – Identification par la méthode de Strejc

au point d'inflexion. On obtient alors les caractéristiques T_1 et T_2 de la réponse. Alors, les paramètres de la fonction de transfert (1) sont déduits du tableau ci-dessous.

n	$\frac{T_1}{\tau}$	$\frac{T_2-T_1}{\tau}$	$\frac{T_1}{T_2-T_1}$
1	0	1	0
2	$0,28$	$2,72$	$0,1$
3	$0,8$	$3,7$	$0,22$
4	$1,42$	$4,46$	$0,32$

TAB. 1 – Paramètres du modèle selon la méthode de Strejc, en fonction de T_1 et T_2

On procède comme suit :

1. tout d'abord on identifie l'ordre du système, d'après la troisième colonne du tableau. Si la valeur de $\frac{T_1}{T_2-T_1}$ ne correspond pas à un entier, ce qui est presque toujours le cas, on choisit pour n l'entier immédiatement inférieur ;
2. une fois n déterminé, on déduit la constante de temps τ du système d'après l'indication de la seconde colonne ;
3. on estime le temps de retard t_r en comparant la valeur de T_1 donnée en appliquant la formule de la première colonne du tableau à la valeur mesurée de T_1 ;
4. K est bien évidemment toujours obtenu directement d'après le régime permanent de la réponse.

C Figures réponses

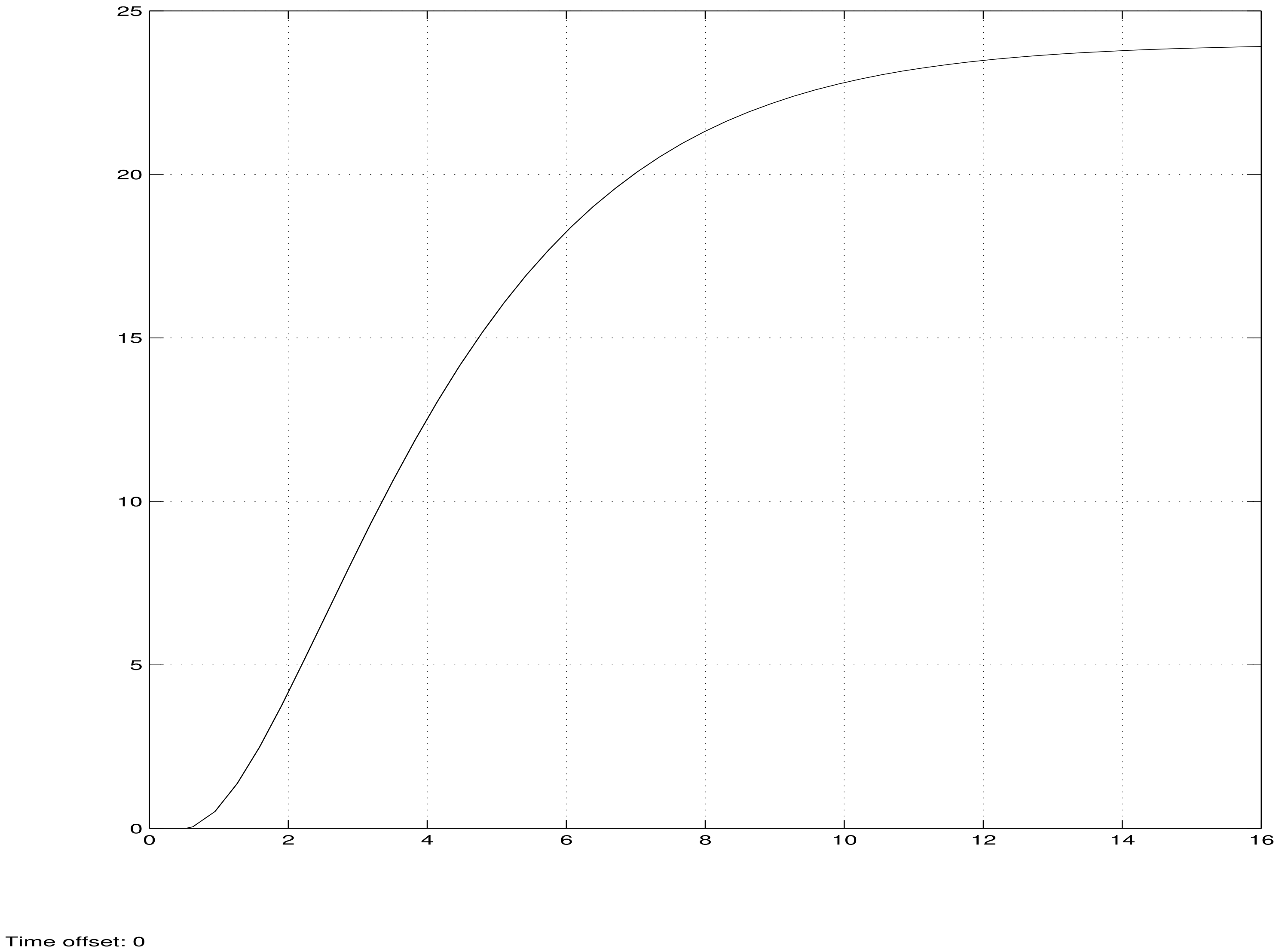


FIG. 5 – Identification du système d'après sa réponse indicielle

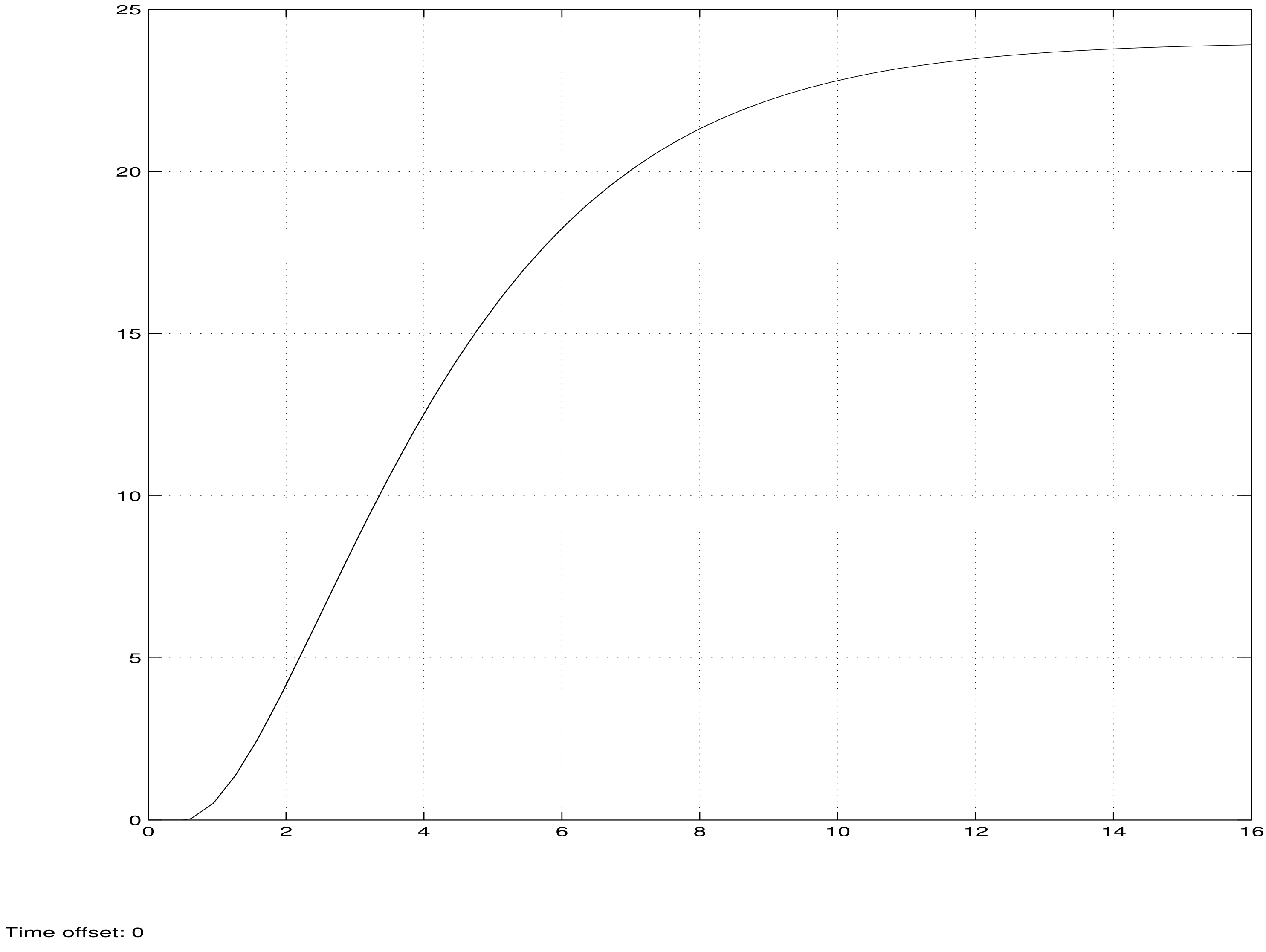


FIG. 6 – Identification du système d'après sa réponse indicielle : méthode de Strejc

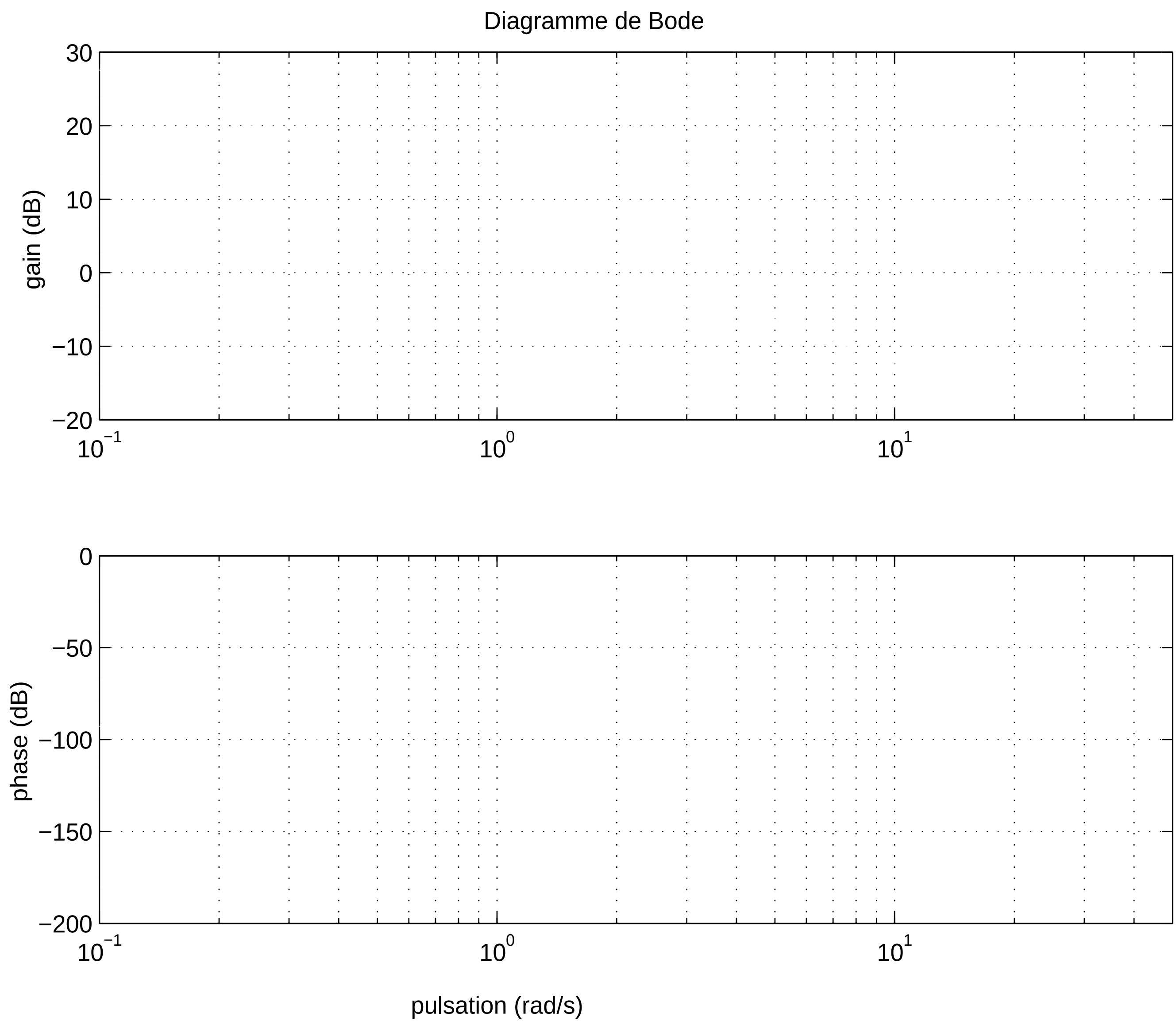


FIG. 7 – Diagramme de Bode