



People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Science and Technology of Oran - Mohamed BOUDIAF
Faculty of Electrical Engineering
Vice-Dean, Postgraduate, Scientific Research and External Relations
وزارة التعليم العالي والبحث العلمي
جامعة وهران للعلوم والتكنولوجيا محمد بوضياف
كلية الهندسة الكهربائية
نيابة عمادة الكلية لما بعد التدرج والبحث العلمي والعلاقات الخارجية



Département d'Électronique

Polycopié Pédagogique

TRAVAUX PRATIQUES DE REGULATION NUMERIQUE

Travaux Pratiques, destinés aux étudiants de :

Deuxième année de Master en Instrumentation

Préparé et présenté par :

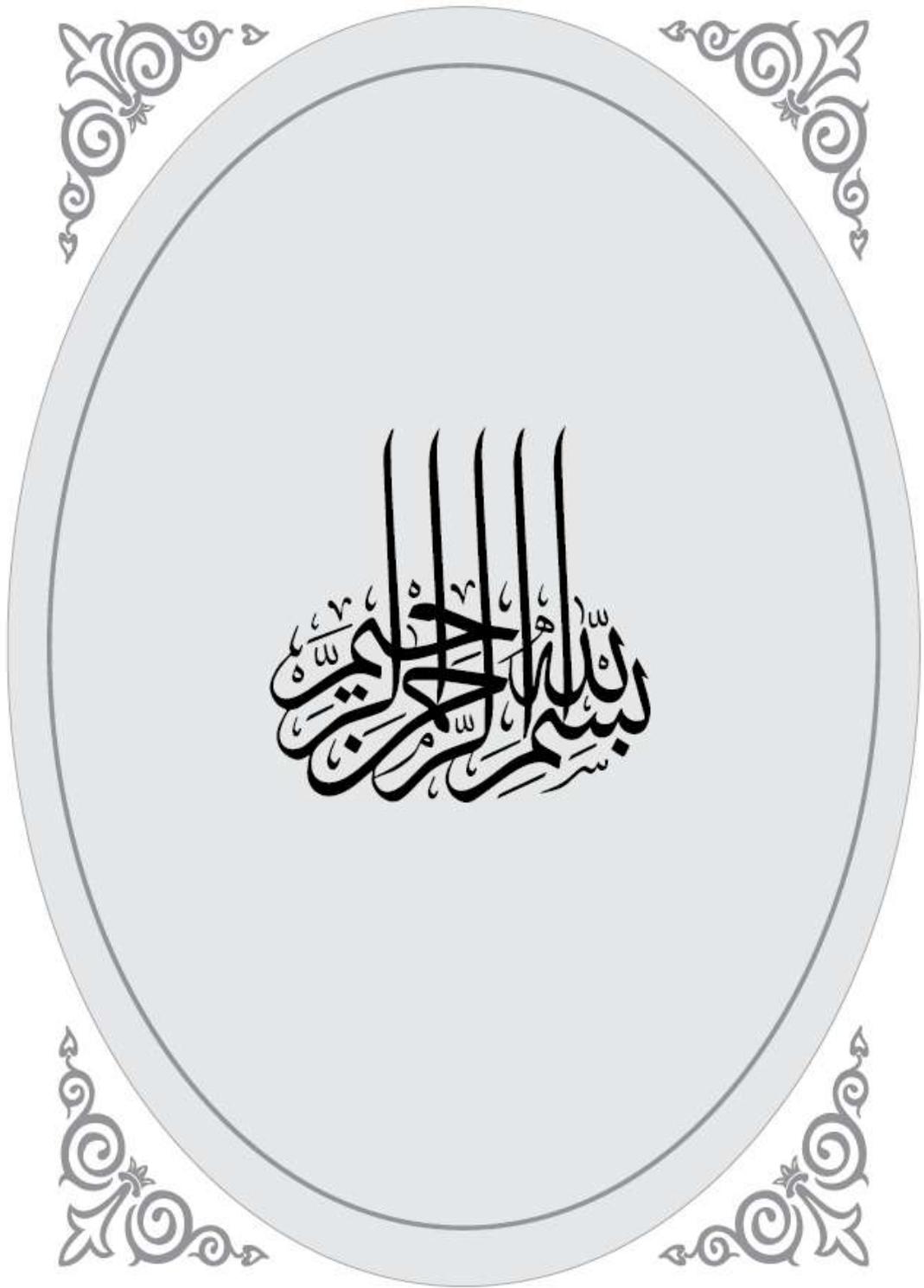
Dr. KHARROUBI Larbi, Maître de conférences A

Comité de lecture :

Dr. LAKHDARI Fethi, Maître de conférences A, USTO-MB

Dr. ZEGRAR Mansour, Maître de conférences A, USTO-MB

Année Universitaire : 2024/2025



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Remerciements

Avant tout, je rends grâce à Allah, Le Très-Miséricordieux, sans la volonté duquel rien n'aurait été possible. Que ces efforts, nourris par Sa sagesse, soient utiles à autrui et Lui soient agréables.

Je remercie chaleureusement Monsieur le Dr. LAKHDARI Fethi, Maitre de conférences A (MCA) à l'université USTO-MB, pour avoir accepté d'expertiser ce polycopié et pour l'attention qu'il a portée à sa lecture.

J'exprime également ma sincère gratitude à Monsieur le Dr. ZEGRAR Mansour, Maitre de conférences A (MCA) à l'université USTO-MB, pour le temps consacré à l'évaluation de ce travail et pour sa bienveillance tout au long du processus.

Table des matières

Table des matières	i
Introduction générale	1
TP N°1 : Modélisation des systèmes LTI discrets à l'aide du logiciel MATLAB	2
Objectif	2
I. Rappels théoriques	2
I.1. Transformée en 'z' d'un signal continu causal	2
I.2. Définitions.....	2
I.3. Représentation des systèmes linéaires LTI discrets monovariante	2
I.4. Discrétisation des systèmes LTI continus monovariante.....	3
I.4.1. Discrétisation par un bloqueur d'ordre zéro (BOZ)	3
I.4.2. Discrétisation par un bloqueur d'ordre un (BOU)	3
I.4.3. Discrétisation par approximation de Tustin	4
I.5. Choix de la période d'échantillonnage	4
I.6. Connexion des systèmes LTI discrets monovariante	5
II. Travail demandé	5
II.1. Préliminaires	5
II.2. Travailler depuis la fenêtre de commande de MATLAB	5
II.2.1. Représentation des systèmes linéaires invariants (LTI) monovariante	5
II.2.1.1. Cas des systèmes LTI continus	5
II.2.1.2. Cas des systèmes LTI discrets	6
II.2.2. Discrétisation des systèmes LTI continus monovariante.....	6
II.2.2.1. Cas d'un système du premier ordre.....	6
II.2.2.2. Cas d'un système du second ordre.....	6
II.2.3. Connexion des systèmes LTI discrets monovariante	7
III. Conclusion.....	8
TP N°2 : Évaluation de stabilité des systèmes LTI discrets à l'aide du logiciel MATLAB.....	9
Objectif	9
I. Rappels théoriques	9
I.1. Évaluation de stabilité par le théorème de stabilité des systèmes LTI discrets	9
I.2. Évaluation de stabilité par la méthode de la réponse impulsionnelle.....	10
II. Travail demandé	11
II.1. Préliminaires	11
II.2. Étude en boucle ouverte (BO)	11
II.2.1. Définition et discrétisation d'un système continu du second ordre.....	11

II.2.2. Évaluation de stabilité en BO	11
II.3. Évaluation de stabilité en boucle fermée (BF)	12
III. Conclusion.....	13
TP N°3 : Analyse temporelle des systèmes LTI discrets à l'aide du logiciel MATLAB...14	
Objectif	14
I. Rappels théoriques	14
I.1. Réponse impulsionnelle	14
I.2. Réponse indicielle	14
I.3. Performances temporelles	14
I.4. Systèmes du premier ordre.....	15
I.5. Systèmes du second ordre.....	16
a. Système à deux pôles réels ($\xi > 1$)	16
b. Système à deux pôles complexes conjugués ($0 < \xi < 1$)	16
c. Système à un pôle réel double ($\xi = 1$)	16
II. Travail demandé	17
II.1. Préliminaires	17
II.2. Réponses et performances temporelles d'un système du premier ordre	18
II.3. Réponses et performances temporelles d'un système du second ordre.....	19
III. Conclusion.....	21
TP N°4 : Analyse fréquentielle des systèmes LTI discrets à l'aide du logiciel MATLAB.....22	
Objectif	22
I. Rappels théoriques	22
I.1. Réponse fréquentielle	22
I.2. Marges de stabilité.....	22
a. Marge de gain ($G_M(\text{dB})$)	22
b. Marge de phase ($\varphi_M(\text{degré})$)	22
II. Travail demandé	24
II.1. Préliminaires	24
II.2. Application.....	24
III. Conclusion.....	25
TP N°5 : Numérisation des contrôleurs PID analogiques réglés par les méthodes de Ziegler-Nichols26	
Objectif	26
I. Rappels théoriques	26
I.1. Les contrôleurs PID analogiques.....	26
I.2. Les contrôleurs PID numériques	27
I.3. Méthodes de réglage de Ziegler-Nichols des contrôleurs PID analogiques	28
II. Travail demandé	31
II.1. Préliminaires	31

II.2. Réglage par l'approche temporelle de Ziegler-Nichols	31
II.3. Réglage par l'approche fréquentielle de Ziegler-Nichols.....	32
III. Conclusion.....	32
TP N°6 : Réglage des contrôleurs PID numériques par la méthode de Takahashi à l'aide du logiciel MATLAB/Simulink.....	33
Objectif	33
I. Rappels théoriques	33
I.1. Méthode de réglage de Takahashi des contrôleurs PID numériques.....	33
a. Essai en boucle ouverte 'BO' (Méthode de la réponse indicielle)	33
b. Essai en boucle fermée 'BF' (Phénomène de pompage)	34
I.2. Les contrôleurs PID numériques	35
I.2.1. Contrôleur P numérique	35
I.2.2. Contrôleur PI numérique	35
a. Forme pédagogique	35
b. Forme industrielle	35
I.2.3. Contrôleur PID numérique	35
a. Forme pédagogique	35
b. Forme industrielle 1 (PID numérique filtré forme 1)	35
c. Forme industrielle 2 (PID numérique filtré forme 2)	36
II. Travail demandé	36
II.1. Préliminaires	36
II.2. Réglage par l'essai en boucle ouverte (BO) de Takahashi.....	36
II.3. Réglage par l'essai en boucle fermée (BF) de Takahashi	38
II.4. Essai en BO de Takahashi VS essai en BF de Takahashi	38
II.4.1. Comparaison des contrôleurs P numériques	38
II.4.2. Comparaison des contrôleurs PI numériques	39
II.4.3. Comparaison des contrôleurs PID numériques	39
III. Conclusion.....	39
TP N°7 : Régulation de la vitesse d'un moteur à courant continu par des contrôleurs PID numériques	40
Objectif	40
I. Rappels théoriques	40
I.1. Modélisation d'un moteur à courant continu (MCC)	40
- Modèle du premier ordre.....	41
- Modèle du second ordre	42
I.2. Synthèse des contrôleurs PID numériques par placement de pôles	42
I.2.1. Commande d'un premier ordre discret par un contrôleur PI numérique	42
- Comportement en premier ordre	43
- Comportement en second ordre.....	43
I.2.2. Commande d'un second ordre discret par un contrôleur PID numérique filtré	44
II. Travail demandé	44
II.1. Préliminaires	44

II.2. Commande PI numérique de la vitesse du moteur décrit par son modèle du premier ordre	44
II.3. Commande PID numérique filtré de la vitesse du moteur décrit par son modèle du second ordre.....	46
III. Conclusion.....	46
TP N°8 : Commande PID numérique sur structure polynomiale RST	47
Objectif	47
I. Rappels théoriques sur les contrôleurs PID numériques filtrés modélisés sous forme polynomiale RST.....	47
II. Travail demandé	48
II.1. Préliminaires	48
II.2. Applications	48
II.2.1. Application 1 : Étude en boucle ouverte.....	48
II.2.2. Application 2 : Étude en boucle fermée	49
II.2.2.1. Commande par un contrôleur PID numérique filtré	49
II.2.2.2. Commande par un contrôleur PID numérique filtré modélisé sous forme RST	50
II.2.2.3. Comparaison entre le PID numérique filtré et le PID modélisé sous forme RST	50
III. Conclusion.....	51
Conclusion générale.....	52
Références	53
Annexe	54
Notations	59
Abréviations	62

Introduction générale

Ce polycopié de Travaux Pratiques (TPs) de Régulation Numérique s'inscrit dans le cadre de Master Instrumentation et s'adresse principalement et plus particulièrement aux étudiants de deuxième année du Département d'Électronique de la Faculté de Génie Électrique (USTO-MB). En fait, ce polycopié est basé sur le matériel pédagogique du cours Éléments de Régulation Numérique.

Ce polycopié expose plusieurs TP de régulation numérique qui se font entièrement sous le logiciel de simulation numérique MATLAB y compris sa boîte à outils "Control System Toolbox" et son environnement graphique "Simulink". Ces TPs portent sur des notions abordées lors des cours d'éléments de régulation numérique. Ils s'articulent sur les systèmes linéaires invariants dans le temps et monovariante, leur modélisation, leur analyse et leur commande par différents contrôleurs, à savoir les PID et les RST.

Ce polycopié est divisé en huit TP :

TP N°1 : Modélisation des systèmes LTI discrets à l'aide du logiciel MATLAB

TP N°2 : Évaluation de stabilité des systèmes LTI discrets à l'aide du logiciel MATLAB

TP N°3 : Analyse temporelle des systèmes LTI discrets à l'aide du logiciel MATLAB

TP N°4 : Analyse fréquentielle des systèmes LTI discrets à l'aide du logiciel MATLAB

TP N°5 : Numérisation des contrôleurs PID analogiques réglés par les méthodes de Ziegler-Nichols

TP N°6 : Réglage des contrôleurs PID numériques par la méthode de Takahashi à l'aide du logiciel MATLAB/Simulink

TP N°7 : Régulation de la vitesse d'un moteur à courant continu par des contrôleurs PID numériques

TP N°8 : Commande PID numérique sur structure polynomiale RST

Objectifs du polycopié

Permettre aux étudiants d'exploiter et de maîtriser les notions théoriques et les phénomènes physiques étudiés en cours.

Connaissances préalables recommandées

- Cours éléments de régulation numérique ;
- Cours systèmes asservis numériques ;
- Cours asservissements et régulation ;
- Logiciel MATLAB/Simulink.

TP N°1
Modélisation des systèmes LTI discrets à l'aide du logiciel MATLAB

Objectif

Savoir modéliser (représenter) les systèmes LTI discrets (échantillonnés) à l'aide de la boîte à outil "Control System Toolbox" du logiciel MATLAB.

I. Rappels théoriques

I.1. Transformée en 'z' d'un signal continu causal

La transformée en 'z' d'un signal continu causal $x(t)$ peut être définie par le diagramme suivant :

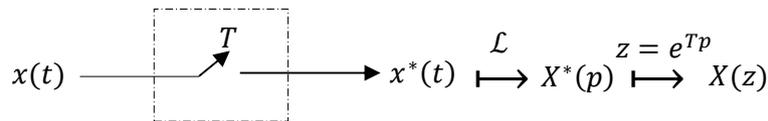


Figure (1) : Diagramme montrant le passage de $x(t)$ à $X(z)$

où :

$x^*(t)$: est le signal échantillonné du signal continu $x(t)$, $X^*(p)$: est la transformée de Laplace de $x^*(t)$, $X(z)$: est la transformée en 'z' du signal continu $x(t)$, telle que :

$$X(z) = \mathcal{Z}[x(t)] = \sum_{k=0}^{\infty} x(kT) z^{-k} \tag{1}$$

Ici, \mathcal{L} et \mathcal{Z} représentent respectivement les opérateurs de Laplace et de la transformée en 'z'.

I.2. Définitions

- Un système est dit linéaire s'il répond au principe de superposition et de proportionnalité.
- Un système est dit discret lorsque les grandeurs le caractérisant génèrent une information à chaque pas de discrétisation.
- Un système est dit linéaire invariant dans le temps (LTI) discret s'il est décrit par des équations récurrentes linéaires à coefficients réels constants.
- Un système est dit monovarié s'il possède un seul signal d'entrée et un seul signal de sortie.
- Une fonction de transfert d'un système LTI discret est la transformée en 'z' de sa réponse impulsionnelle.

I.3. Représentation des systèmes linéaires LTI discrets monovarié

Un système linéaire invariant dans le temps LTI discret peut être décrit au moyen des quatre façons suivantes :

- une fonction de transfert discrète sous forme polynomiale en 'z' ou bien en 'z⁻¹' ;
- une fonction de transfert discrète sous forme zéro-pôle et gain (forme d'Evans) en 'z' ou en 'z⁻¹' ;
- des équations récurrentes (équations aux différences) ;
- des modèles d'état linéaires discrets (représentation avancée).

I.4. Discrétisation des systèmes LTI continus monovariante

Soit un système LTI continu monovariante défini par une fonction de transfert continue $G(p)$, telle que :

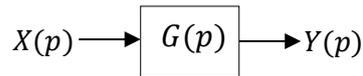


Figure (2) : Schéma fonctionnel d'un système LTI continu monovariante

Plusieurs méthodes de discrétisation peuvent être utilisées afin de numériser le système $G(p)$. Dans cette section, on va citer les principales méthodes. On définit $G(z)$ est la fonction de transfert discrète du système $G(p)$.

I.4.1. Discrétisation par un bloqueur d'ordre zéro (BOZ)

En utilisant un bloqueur d'ordre zéro (BOZ), la fonction de transfert discrète $G(z)$ est obtenue comme suit :

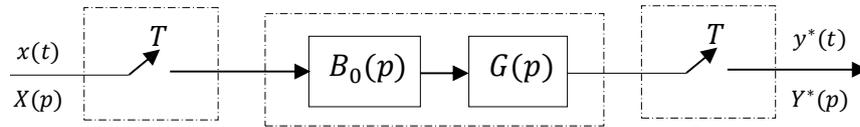


Figure (3) : Schéma fonctionnel figurant la discrétisation du système $G(p)$ continu par un BOZ

↔

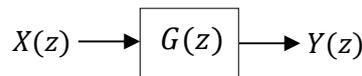


Figure (4) : Schéma fonctionnel d'un système LTI discret monovariante correspondant

d'où :

$$G(z) = (1 - z^{-1})Z \left[\frac{G(p)}{p} \right] \quad (2)$$

avec $B_0(p)$ est la fonction de transfert d'un BOZ de période d'échantillonnage T , telle que :

$$B_0(p) = \frac{1 - e^{-Tp}}{p} \quad (3)$$

I.4.2. Discrétisation par un bloqueur d'ordre un (BOU)

Par l'utilisation d'un bloqueur d'ordre un (BOU), la fonction $G(z)$ est obtenue comme suit :

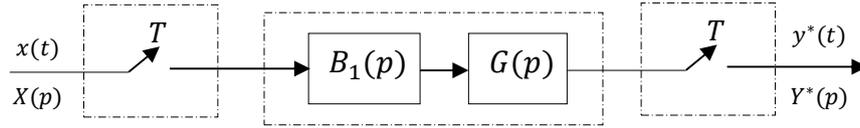


Figure (5) : Schéma fonctionnel figurant la discrétisation du système $G(p)$ continu par un BOU

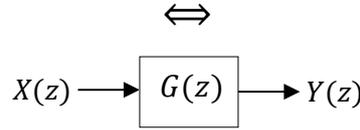


Figure (6) : Schéma fonctionnel d'un système LTI discret monovariante correspondant

d'où :

$$G(z) = (1 - z^{-1})^2 Z \left[\frac{1+Tp}{Tp^2} G(p) \right] \quad (4)$$

La fonction de transfert du bloqueur d'ordre un (BOU) est définie par:

$$B_1(p) = (1 - e^{-Tp})^2 \left(\frac{1+Tp}{Tp^2} \right) \quad (5)$$

I.4.3. Discrétisation par approximation de Tustin

Il est également possible d'obtenir la fonction $G(z)$ en utilisant l'approximation de 'Tustin' (approximation bilinéaire) :

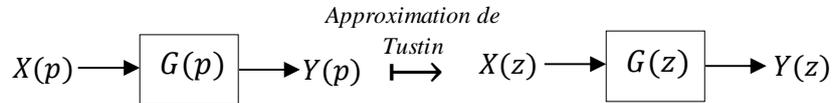


Figure (7) : Schéma fonctionnel figurant la discrétisation du système $G(p)$ continu par approximation de Tustin

d'où :

$$G(z) = G(p) \Big|_{p=\frac{2}{T} \cdot \frac{z-1}{z+1}} \quad (6)$$

I.5. Choix de la période d'échantillonnage

La période d'échantillonnage T peut être choisie comme suit :

- pour un système du premier ordre : $0.25\tau \leq T \leq 1.25\tau$ (7)

- pour un système du second ordre : $\frac{0.25}{\omega_0} \leq T \leq \frac{1.25}{\omega_0}$ (8)

- pour un système d'ordre quelconque : $5f_c \leq f \leq 25f_c \Leftrightarrow 2\pi/25\omega_c \leq T \leq 2\pi/5\omega_c$ (9)

où f , f_c , et ω_c sont respectivement la fréquence d'échantillonnage, la fréquence de coupure, et la pulsation de coupure du système à discrétiser. τ est la constante de temps du système du premier ordre à discrétiser et ω_0 est la pulsation propre du système du second ordre à discrétiser.

I.6. Connexion des systèmes LTI discrets monovariante

Dans le domaine discret, les systèmes LTI peuvent être mis en série, en parallèle ou en rétroaction (feedback) :

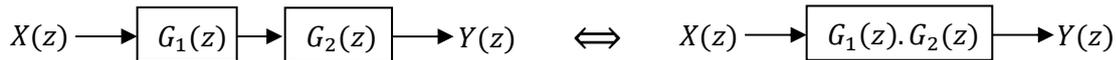


Figure (8) : Deux systèmes LTI discrets montés en série

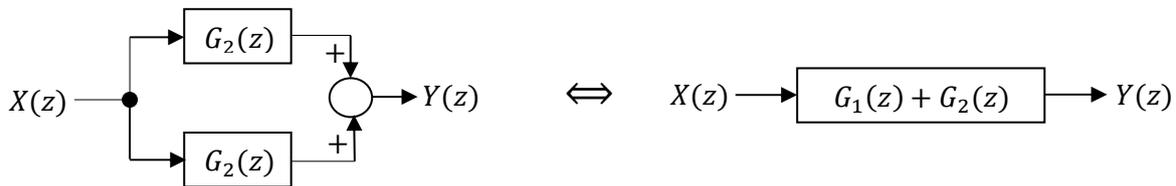


Figure (9) : Deux systèmes LTI discrets montés en parallèle

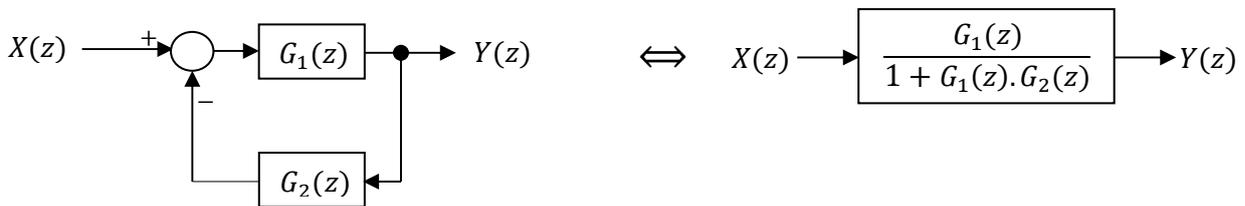


Figure (10) : Deux systèmes LTI discrets montés en rétroaction (feedback)

II. Travail demandé

II.1. Préliminaires

1. Lancer MATLAB.
2. Afficher le bureau par défaut de MATLAB.
3. Créer votre dossier personnel correspondant au premier TP dans le répertoire C:\Documents\MATLAB portant le nom TPN01.
4. Ajouter votre dossier personnel au Path de MATLAB.

II.2. Travailler depuis la fenêtre de commande de MATLAB

II.2.1. Représentation des systèmes linéaires invariants (LTI) monovariante

II.2.1.1. Cas des systèmes LTI continus

En utilisant la boîte à outil "Control System Toolbox" :

- Créer le système continu suivant:

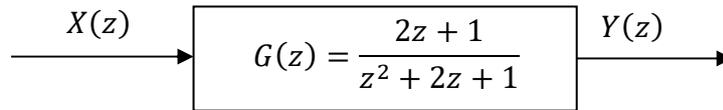
$$G(p) = \frac{2p + 1}{p^2 + 2p + 1}$$

- Montrer que la fonction de transfert $G(p)$ peut se mettre sous la forme:

$$G(p) = 2 \cdot \frac{p + 0.5}{(p + 1)^2}$$

II.2.1.2. Cas des systèmes LTI discrets

Soit un système LTI discret décrit par la fonction de transfert discrète suivante:



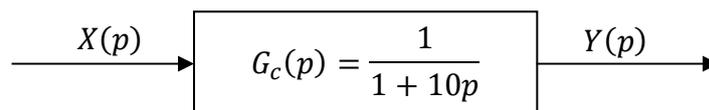
En utilisant la boîte à outil "Control System Toolbox" :

1. Créer la fonction de transfert discrète $G(z)$.
2. Réécrire la fonction de transfert discrète $G(z)$ en fonction de z^{-1} .
3. Réécrire la fonction de transfert discrète $G(z)$ sous forme zéro-pôle et gain (forme d'Evans).
4. Déterminer les pôles, les zéros et le gain d'Evans de $G(z)$.

On donne comme période d'échantillonnage : $T = 1$ (seconde).

II.2.2. Discrétisation des systèmes LTI continus monovariante

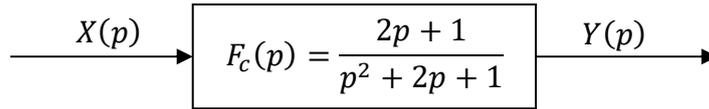
II.2.2.1. Cas d'un système du premier ordre



1. Définir la fonction de transfert continue $G_c(p)$.
2. Calculer $T_{Min} = 0.25\tau$ et $T_{Max} = 1.25\tau$.
3. Choisir une période d'échantillonnage T dans la fourchette : $T_{Min} \leq T \leq T_{Max}$. Puis discrétiser le système continu $G_c(p)$ par les méthodes suivantes :
 - Discrétisation par un bloqueur d'ordre zéro 'BOZ'.
 - Discrétisation par un bloqueur d'ordre un 'BOU'.
 - Discrétisation par l'approximation de 'Tustin'.
4. Vérifier les résultats obtenus par MATLAB avec ceux obtenus par le calcul analytique.

II.2.2.2. Cas d'un système du second ordre

Soit la fonction de transfert continue suivante:



1. Définir la fonction de transfert continue $F_c(p)$.
2. Tracer le diagramme de Bode en boucle ouverte du système $F_c(p)$.
3. Depuis le diagramme de gain de Bode, déterminer la pulsation de coupure ω_c du système $F_c(p)$.
4. Calculer $T_{Min} = 2\pi/25\omega_c$ et $T_{Max} = 2\pi/5\omega_c$.
5. Choisir une période d'échantillonnage T dans la fourchette : $T_{Min} \leq T \leq T_{Max}$. Puis, discrétiser le système continu $F_c(p)$ par les méthodes suivantes :

- Discrétisation par un bloqueur d'ordre zéro 'BOZ'.
- Discrétisation par un bloqueur d'ordre un 'BOU'.
- Discrétisation par l'approximation de 'Tustin'.

6. Vérifier les résultats obtenus par MATLAB avec ceux obtenus par le calcul analytique.

II.2.3. Connexion des systèmes LTI discrets monovariante

On considère les deux fonctions discrètes suivantes :

$$G_1(z) = \frac{1}{z+1} \quad \text{et} \quad G_2(z) = \frac{1}{z+2}$$

1. En utilisant la boîte à outils "Control System Toolbox", créer les fonctions de transfert discrètes $G_s(z)$, $G_p(z)$, et $G_f(z)$, telles que:

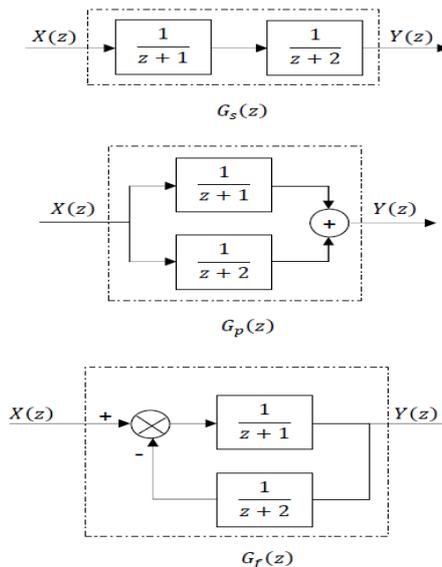


Figure (11) : Deux systèmes LTI discrets montés en différentes situations

2. Vérifier les résultats obtenus par MATLAB avec ceux obtenus par le calcul analytique.

III. Conclusion

Ce premier TP introduit la modélisation des systèmes LTI discrets sous MATLAB, en utilisant différentes méthodes de discrétisation (BOZ, BOU, Tustin). Il permet de représenter ces systèmes sous diverses formes et d'étudier leurs interconnexions. L'influence de la période d'échantillonnage sur le comportement dynamique est également analysée.

TP N°2

Évaluation de stabilité des systèmes LTI discrets à l'aide du logiciel MATLAB

Objectif

Dans ce TP, nous allons utiliser le logiciel MATLAB pour évaluer la stabilité, en boucle ouverte (BO) et en boucle fermée (BF), d'un système LTI discret du second ordre. Pour ce faire, deux méthodes sont considérées : le théorème de stabilité des systèmes LTI discrets et la méthode de la réponse impulsionnelle.

I. Rappels théoriques

I.1. Évaluation de stabilité par le théorème de stabilité des systèmes LTI discrets

Pour qu'un système LTI discret soit stable en boucle fermée (ou en boucle ouverte), il faut que toutes les racines de l'équation caractéristique de sa fonction de transfert en boucle fermée (ou en boucle ouverte) soient à l'intérieur d'un cercle de rayon $R_0 = 1$ (cercle unité) et de centre (0,0) dans le plan complexe 'z', comme le montre la figure suivante :

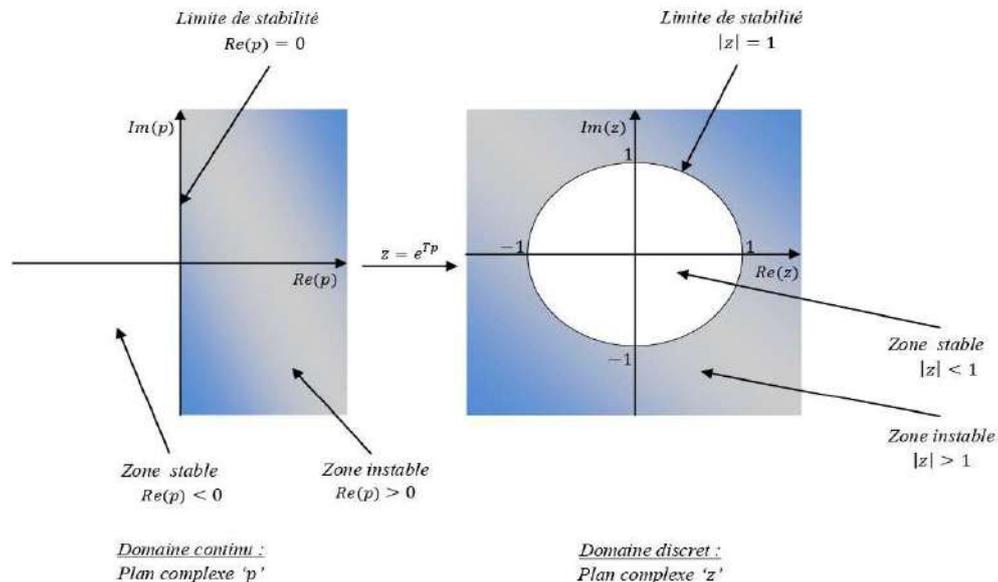


Figure (1) : Correspondance entre le plan complexe 'p' et le plan complexe 'z' :
Passage du domaine continu au domaine discret

Soit le système discret décrit par la fonction de transfert :

$$G(z) = \frac{N(z)}{D(z)} = \frac{b_0 z^m + b_1 z^{m-1} + \dots + b_m}{a_0 z^n + a_1 z^{n-1} + \dots + a_n} \quad (1)$$

On définit les pôles p_i du système $G(z)$ qui sont les racines de l'équation (polynôme) caractéristique :

$$D(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_n = 0 \quad (2)$$

alors :

$$\text{Si } \forall i, |p_i| < 1 \Leftrightarrow \text{Le système } G(z) \text{ est stable} \quad (3)$$

I.2. Évaluation de stabilité par la méthode de la réponse impulsionnelle

Réponse impulsionnelle : est la réponse de sortie d'un système résultante d'une entrée en impulsion de Dirac (Voir Figure (2)).

La réponse impulsionnelle $y(k)$ d'un système LTI discret est donnée par la convolution numérique suivante :

$$y(k) = \sum_{i=0}^{+\infty} g(k-i)\delta(i) = g(k), k \geq 0 \quad (4)$$

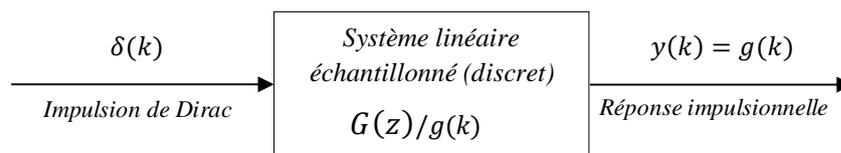


Figure (2) : Réponse impulsionnelle d'un système linéaire échantillonné (discret)

Un système LTI discret est stable si sa réponse impulsionnelle converge vers zéro :

$$\text{Si } \lim_{k \rightarrow \infty} y(k) = 0 \Leftrightarrow \text{Le système } G(z) \text{ est stable} \quad (5)$$

La figure de MATLAB suivante montre la stabilité d'un système LTI discret via sa réponse impulsionnelle :

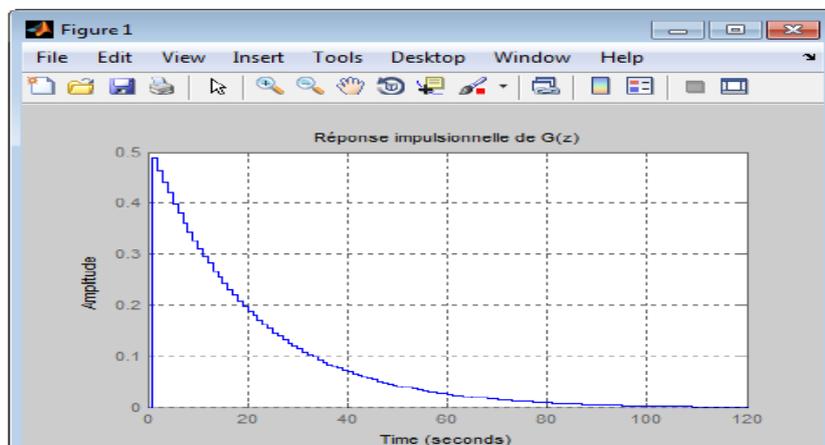


Figure (3) : Système LTI discret stable : Réponse impulsionnelle converge vers zéro

II. Travail demandé

II.1. Préliminaires

1. Lancer MATLAB.
2. Afficher le bureau par défaut de MATLAB.
3. Créer votre dossier personnel correspondant au deuxième TP dans le répertoire C:\Documents\MATLAB portant le nom TPN02.
4. Ajouter votre dossier personnel au Path de MATLAB.

II.2. Étude en boucle ouverte (BO)

II.2.1. Définition et discrétisation d'un système continu du second ordre

En utilisant un fichier *script (M-file)*, faire :

1. Définir le système continu du second ordre suivant: $F_c(p) = \frac{1}{p^2+p+1}$.
2. Tracer son diagramme de Bode en boucle ouverte.
3. Depuis le diagramme de gain de Bode, déduire la pulsation de coupure ' ω_c ' de $F_c(p)$.
4. Calculer $T_{Min} = 2\pi/(25\omega_c)$ et $T_{Max} = 2\pi/(5\omega_c)$.
5. Discrétiser la fonction continue $F_c(p)$, en utilisant la méthode de 'BOZ', avec les différentes valeurs de T suivantes : $T_1 = T_{Min}$, $T_2 = (T_{Max} + T_{Min})/2$ et $T_3 = T_{Max}$. Conclure.

On définit : F_{d1} , F_{d2} et F_{d3} sont respectivement les fonctions de transfert obtenues avec T_1 , T_2 et T_3 .

II.2.2. Évaluation de stabilité en BO

- Évaluer la stabilité en boucle ouverte de : F_{d1} , F_{d2} et F_{d3} , en utilisant :
 - le théorème de stabilité des systèmes LTI discrets ;
 - la méthode de la réponse impulsionnelle.
- Puis, remplir les tableaux suivants :

	<i>Le système discret F en BO est-il stable ?</i>
<i>Le système F décrit par $F_{d1}(z)$</i>	
<i>Le système F décrit par $F_{d2}(z)$</i>	
<i>Le système F décrit par $F_{d3}(z)$</i>	

Tableau (1) : Évaluation de stabilité en BO du système discret F par le théorème de stabilité des systèmes LTI discrets

	<i>Le système discret F en BO est-il stable ?</i>
<i>Le système F décrit par $F_{d1}(z)$</i>	
<i>Le système F décrit par $F_{d2}(z)$</i>	
<i>Le système F décrit par $F_{d3}(z)$</i>	

Tableau (2) : Évaluation de stabilité en BO du système discret F via sa réponse impulsionnelle

II.3. Évaluation de stabilité en boucle fermée (BF)

On considère le système bouclé suivant :

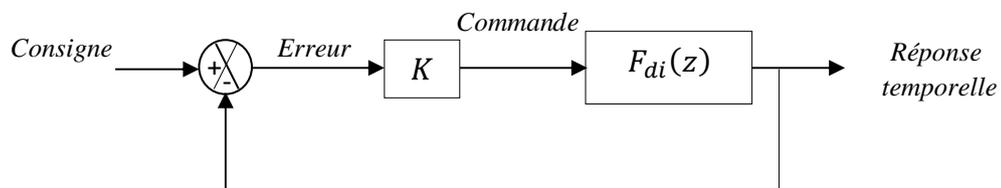


Figure (3) : Le système discret F dans une boucle fermée

En utilisant le même fichier script (*M-file*), faire :

a. En fixant le paramètre K par 1, évaluer la stabilité en boucle fermée, de : F_{d1} , F_{d2} et F_{d3} , en utilisant :

- le théorème de stabilité des systèmes LTI discrets ;
- la méthode de la réponse impulsionnelle.

- Puis, remplir les tableaux suivants :

	<i>Le système discret F en BF est-il stable ?</i>
<i>Le système F décrit par $F_{d1}(z)$</i>	
<i>Le système F décrit par $F_{d2}(z)$</i>	
<i>Le système F décrit par $F_{d3}(z)$</i>	

Tableau (3) : Évaluation de stabilité en BF du système discret F par le théorème de stabilité des systèmes LTI discrets

	<i>Le système discret F en BF est-il stable ?</i>
<i>Le système F décrit par $F_{d1}(z)$</i>	
<i>Le système F décrit par $F_{d2}(z)$</i>	
<i>Le système F décrit par $F_{d3}(z)$</i>	

Tableau (4) : Évaluation de stabilité en BF du système discret F via sa réponse impulsionnelle

b. Refaire la question **a** en ajustant le paramètre K . Puis, trouver le domaine de stabilité du système bouclé en fonction du paramètre K .

III. Conclusion

Ce deuxième TP vise à évaluer la stabilité des systèmes LTI discrets en boucle ouverte et fermée, à travers l'analyse de la réponse impulsionnelle et l'application du théorème de stabilité. L'influence de la période d'échantillonnage sur la stabilité y est mise en évidence, ainsi que les conditions nécessaires pour assurer un comportement stable.

TP N°3

Analyse temporelle des systèmes LTI discrets à l'aide du logiciel MATLAB

Objectif

Savoir-faire l'analyse temporelle des systèmes LTI discrets à l'aide de la boîte à outil "Control System Toolbox" du logiciel MATLAB.

I. Rappels théoriques

I.1. Réponse impulsionnelle : est la réponse de sortie d'un système résultant d'une entrée en impulsion de Dirac.

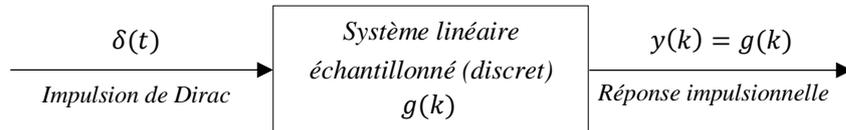


Figure (1) : Réponse impulsionnelle d'un système linéaire échantillonné (discret)

I.2. Réponse indicielle : est la réponse de sortie d'un système résultant d'une entrée en échelon unité.

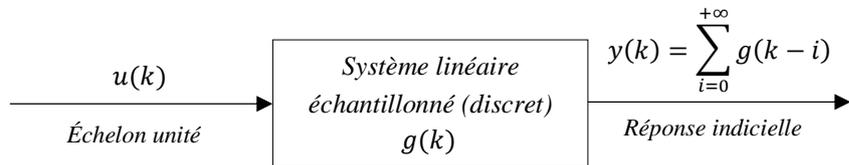


Figure (2) : Réponse indicielle d'un système linéaire échantillonné (discret)

I.3. Performances temporelles

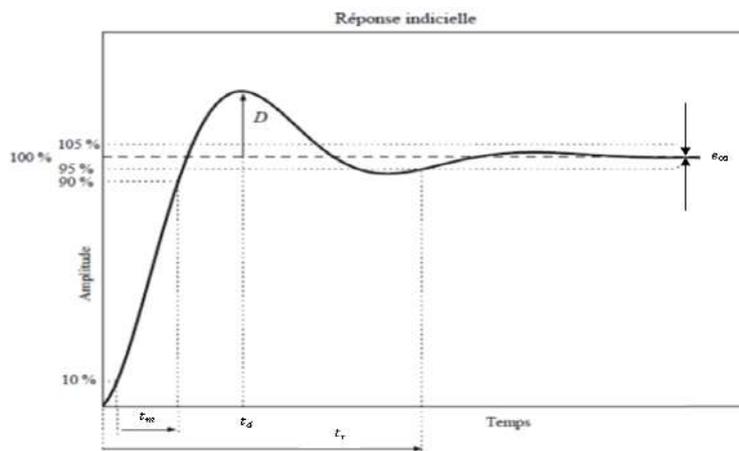


Figure (3) : Performances temporelles associées à une réponse indicielle

Cette réponse est caractérisée par les performances temporelles suivantes:

- un premier dépassement (le premier pic) D : traduisant le degré d'amortissement (le degré de stabilité) du système ;
- le temps de premier pic t_d : traduisant la rapidité du régime transitoire ;
- le temps de montée t_m : comme étant le temps que met la réponse indicielle pour aller de **10%** à **90%** de sa valeur finale y_∞ . Ce temps traduisant la rapidité du régime transitoire ;
- le temps de réponse t_r : c'est le temps requis (nécessaire) pour que la courbe de sortie atteigne et reste à l'intérieur d'une bande, exprimée en pourcentage (**2%** ou **5%**.), relativement à sa valeur finale y_∞ . Il permet de mesurer la rapidité du système asservi ;
- l'erreur statique e_∞ : l'erreur statique du système est caractérisée par l'erreur en régime permanent en présence d'un échelon. Elle permet de mesurer la précision du système asservi.

I.4. Systèmes du premier ordre

Un système LTI continu du premier ordre est modélisé par une fonction de transfert sous forme normalisée suivante :

$$G(p) = \frac{K}{1+\tau p} \quad (1)$$

où

K et τ sont respectivement le gain statique et la constante du temps du système $G(p)$.

Réponse impulsionnelle du système $G(p)$:

$$y(t) = \frac{K}{\tau} e^{-\frac{t}{\tau}}. u(t) \quad (2)$$

Réponse indicielle du système $G(p)$:

$$y(t) = K \left(1 - e^{-\frac{t}{\tau}}\right). u(t) \quad (3)$$

On définit le système discret du premier ordre $G(z)$ correspondant au système continu $G(p)$ obtenu avec un BOZ de période d'échantillonnage ' T ', dont la fonction de transfert est donnée comme suit :

$$G(z) = \frac{K(1-e^{-\frac{T}{\tau}})}{z-e^{-\frac{T}{\tau}}} \quad (4)$$

Réponse impulsionnelle du système discret $G(z)$:

$$y(k) = K(1 - e^{-\frac{T}{\tau}}) e^{-\frac{(k-1)T}{\tau}}. u(k - 1) \quad (5)$$

Réponse indicielle du système discret $G(z)$:

$$y(k) = K \left(1 - e^{-\frac{kT}{\tau}}\right). u(k) \quad (6)$$

Le temps de réponse t_r :

$$t_r = -\tau \ln(0,05) \approx 3\tau \quad (7)$$

Le temps de montée t_m :

$$t_m = 2.2\tau \quad (8)$$

I.5. Systèmes du second ordre

Un système LTI continu du second ordre sans zéros est caractérisé par la fonction de transfert normalisée suivante:

$$G(p) = \frac{K}{\frac{1}{\omega_0^2}p^2 + \frac{2\xi}{\omega_0}p + 1} \quad (9)$$

avec K son gain statique, ω_0 sa pulsation propre non amortie et ξ son coefficient d'amortissement.

Différents cas sont alors possibles :

a. Système à deux pôles réels ($\xi > 1$)

La fonction de transfert discrète correspondante est donnée par :

$$G(z) = \frac{Az+B}{(z-e^{Tp_1})(z-e^{Tp_2})} \quad (10)$$

où p_1 et p_2 sont les pôles continus et les pôles discrets sont:

$$p_{d1,2} = \omega_0(-\xi \pm \sqrt{\xi^2 - 1}) \quad (11)$$

b. Système à deux pôles complexes conjugués ($0 < \xi < 1$)

On a comme fonction de transfert discrète correspondante:

$$G(z) = \frac{Az+B}{(z-z_1)(z-z_1^*)} \quad (12)$$

Les pôles de cette fonction sont définis par : $z_1 = e^{p_1 T}$ et $z_1^* = e^{p_1^* T}$. Les coefficients A et B sont définis comme suit :

$$\begin{aligned} A &= 1 - e^{-rT} \left(\cos(cT) + \frac{r}{c} \sin(cT) \right) \\ B &= e^{-2rT} + e^{-rT} \left(\frac{r}{c} \sin(cT) - \cos(cT) \right) \end{aligned} \quad (13)$$

avec $p_1 = -\xi\omega_0 + j\omega_0\sqrt{\xi^2 - 1}$, $z_1 = e^{Tp_1}$, p_1^* et z_1^* sont les conjugués de p_1 et z_1 , respectivement. $p_1 = -r + jc$, $p_1^* = -r - jc$, $r = \xi\omega_0$ et $c = \omega_0\sqrt{1 - \xi^2}$.

c. Système à un pôle réel double ($\xi = 1$)

Dans ce cas, la fonction de transfert discrète correspondante est donnée comme suit:

$$G(z) = \frac{A_2 z + B_2}{(z - e^{T\omega_0})^2} \quad (14)$$

avec

$$\begin{aligned} A_2 &= K e^{T\omega_0} (e^{-T\omega_0} + \omega_0 - 1) \\ B_2 &= K e^{T\omega_0} (e^{T\omega_0} + T\omega_0 - 1) \end{aligned} \quad (15)$$

Dans cette section, nous étudions le cas le plus pratique : $0 < \xi < 1$.

La réponse indicielle est exprimée par :

- dans le cas continu

$$y(t) = 1 - \frac{e^{-rt}}{\sqrt{1-\xi^2}} \cos(ct - \beta) \cdot u(t) \quad (16)$$

- dans le cas discret

$$y(kT) = 1 - \frac{e^{-rkT}}{\sqrt{1-\xi^2}} \cos(ckT - \beta) \cdot u(kT) \quad (17)$$

avec

$$\tan(\beta) = \frac{\xi}{\sqrt{1-\xi^2}} \quad (18)$$

D'autre part, on définit les performances temporelles suivantes :

- Le premier dépassement indiciel D est donné par :

$$D = \exp\left(-\frac{\pi\xi}{\sqrt{1-\xi^2}}\right) \quad (19)$$

- Le temps du premier maximum t_d est donné par :

$$t_d = \exp\left(\frac{\pi}{\omega_0\sqrt{1-\xi^2}}\right) = \exp\left(\frac{\pi}{c}\right) \quad (20)$$

- Le temps de montée t_m est donné par :

$$t_m = \frac{1}{\omega_0\sqrt{1-\xi^2}} (\pi - \cos^{-1}(\xi)) \quad (21)$$

- Le temps de réponse t_r à $n\%$ est donné par :

$$t_r = \frac{1}{\omega_0\xi} \ln\left(\frac{100}{n}\right) \quad (22)$$

II. Travail demandé

II.1. Préliminaires

1. Lancer MATLAB.

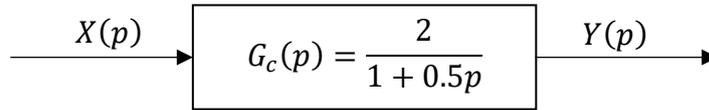
2. Afficher le bureau par défaut de MATLAB.

3. Créer votre dossier personnel correspondant au troisième TP dans le répertoire C:\Documents\MATLAB portant le nom TPN03.

4. Ajouter votre dossier personnel au Path de MATLAB.

II.2. Réponses et performances temporelles d'un système du premier ordre

Soit un système LTI continu du premier ordre dont la fonction de transfert est définie comme suit :



En utilisant un fichier *script* :

1. Définir la fonction de transfert continue $G_c(p)$.
2. Déduire la pulsation de coupure ω_c depuis le diagramme Bode en boucle ouverte du système $G_c(p)$. Puis, calculer T_{Min} et T_{Max} .
3. Discrétiser la fonction continue $G_c(p)$ par un bloqueur d'ordre zéro, avec les différentes valeurs de la période d'échantillonnage T suivantes : $0.5T_{Min}$, T_{Min} , $2T_{Min}$ et T_{Max} .
4. Tracer la réponse impulsionnelle en boucle ouverte 'BO' du système continu $G_c(p)$.
5. Dans la même figure de MATLAB tracer les réponses impulsionnelles en BO du système discret pour les différentes valeurs de T . Commenter.
6. Déterminer graphiquement : le premier dépassement (Peak Response) et le temps de réponse à 5% (Settling Time). Puis remplir le tableau suivant :

	Premier dépassement	Temps de réponse à 5%
Cas du système continu $G_c(p)$		
Cas du système discret pour la valeur $0.5T_{Min}$		
Cas du système discret pour la valeur T_{Min}		
Cas du système discret pour la valeur $2T_{Min}$		
Cas du système discret pour la valeur T_{Max}		

Tableau (1) : Performances temporelles du système du premier ordre obtenues avec les différentes valeurs de T : Cas d'une réponse impulsionnelle

7. Comparer les résultats puis conclure.

8. Créer une autre fenêtre graphique de MATLAB, tracer la réponse indicielle en *BO* du système continu $G_c(p)$.

9. Dans la même figure, tracer les réponses indiciaires en *BO* du système discret pour les différentes valeurs de T . Commenter les résultats.

10. Déterminer graphiquement :

- le premier dépassement (Peak Response)
- le temps de réponse à 5% (SettlingTime).
- le temps de montée (Rise Time).
- l'erreur statique ou bien la valeur finale (Steady State error).

Puis remplir le tableau suivant :

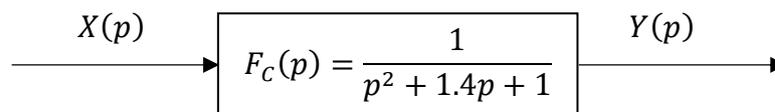
	Premier dépassement	Temps de montée	Temps de réponse à 5%	Erreur statique
Cas du système continu $G_c(p)$				
Cas du système discret pour la valeur $0.5T_{Min}$				
Cas du système discret pour la valeur T_{Min}				
Cas du système discret pour la valeur $2T_{Min}$				
Cas du système discret pour la valeur T_{Max}				

Tableau (2) : Performances temporelles du système du premier ordre obtenues avec les différentes valeurs de T : Cas d'une réponse indicielle

11. Comparer les résultats puis conclure.

II.3. Réponses et performances temporelles d'un système du second ordre

On considère un système LTI continu du second ordre suivant :



En utilisant un autre fichier *script* :

1. Définir la fonction continue $F_c(p)$.
2. Tracer son diagramme de Bode en boucle ouverte.
3. Déduire sa pulsation de coupure ω_c .
4. Calculer T_{Min} et T_{Max} pour le calcul de la période d'échantillonnage T .
5. Discrétiser la fonction continue $F_c(p)$ par un bloqueur d'ordre zéro, avec les différentes valeurs de T suivantes : $0.5T_{Min}$, T_{Min} , $2T_{Min}$ et T_{Max} .
6. En utilisant l'outil 'LTI Viewer' de la boîte à outils "Control System Toolbox" :
 - 6.1. Visualiser à la fois les réponses impulsionnelles en BO :
 - du système continu $F_c(p)$;
 - du système discret décrit par les fonctions de transfert associées avec les différentes valeurs de T .
 - depuis les différentes réponses obtenues, déterminer graphiquement les performances temporelles, puis remplir le tableau (3).
 - Puis comparer et conclure.
 - 6.2. Visualiser à la fois les réponses indicielles en BO :
 - du système continu $F_c(p)$.
 - du système discret décrit par les fonctions de transfert associées avec les différentes valeurs de T .
 - depuis les différentes réponses obtenues, déterminer graphiquement les performances temporelles, remplir le tableau (4), puis comparer et conclure.

	Premier dépassement	Temps de réponse à 5%
Cas du système continu $F_c(p)$		
Cas du système discret pour la valeur $0.5T_{Min}$		
Cas du système discret pour la valeur T_{Min}		
Cas du système discret pour la valeur $2T_{Min}$		
Cas du système discret pour la valeur T_{Max}		

Tableau (3) : Performances temporelles du système du second ordre obtenues avec les différentes valeurs de T : Cas d'une réponse impulsionnelle

	Premier dépassement	Temps de montée	Temps de réponse à 5%	Erreur statique
Cas du système continu $F_c(p)$				
Cas du système discret pour la valeur $0.5T_{Min}$				
Cas du système discret pour la valeur T_{Min}				
Cas du système discret pour la valeur $2T_{Min}$				
Cas du système discret pour la valeur T_{Max}				

Tableau (4) : Performances temporelles du système du second ordre obtenues avec les différentes valeurs de T : Cas d'une réponse indicielle

III. Conclusion

Ce TP met en évidence l'analyse des réponses temporelles (impulsionnelle et indicielle) de systèmes discrets de référence (premier et second ordre). Les performances dynamiques, telles que le temps de réponse, le temps de montée, le premier dépassement et l'erreur statique, sont évaluées, mettant en lumière l'impact des méthodes de discrétisation ainsi que celui de la période d'échantillonnage.

TP N°4

Analyse fréquentielle des systèmes LTI discrets à l'aide du logiciel MATLAB

Objectif

Savoir-faire l'analyse fréquentielle des systèmes LTI discrets à l'aide du logiciel MATLAB.

I. Rappels théoriques

I.1. Réponse fréquentielle

La réponse fréquentielle d'un système LTI discret défini dans le domaine fréquentiel par une fonction de transfert $G(e^{j\omega T})$ est caractérisée par deux paramètres, qui sont :

- le gain : $G_{dB} = |G(e^{j\omega T})|$;
- la phase : $\varphi = \arg(G(e^{j\omega T}))$.

L'étude d'une réponse fréquentielle d'un système discret peut être effectuée soit par :

- le diagramme de Bode ;
- le lieu de Nyquist ;
- le lieu de Black-Nichols.

I.2. Marges de stabilité

La marge de gain et la marge de phase sont des indicateurs qui permettent de quantifier la robustesse de la stabilité vis-à-vis de modifications du processus.

a. Marge de gain (G_M (dB)) : est la différence entre 0 dB et la valeur du gain pour lequel la phase est égale à -180°

b. Marge de phase (φ_M (degré)) : est la différence entre la valeur -180° et la phase pour laquelle le gain est égal à 0 dB.

Les valeurs usuelles de marge de gain et de phase sont :

- Marge de gain : 10 à 12 dB.
- Marge de phase : 45° à 50° .

Les figures (1) et (2) montrent respectivement les marges de gain et de phase associées au diagramme de Bode et au lieu de Nyquist, d'un système discret, obtenues en boucle ouverte:

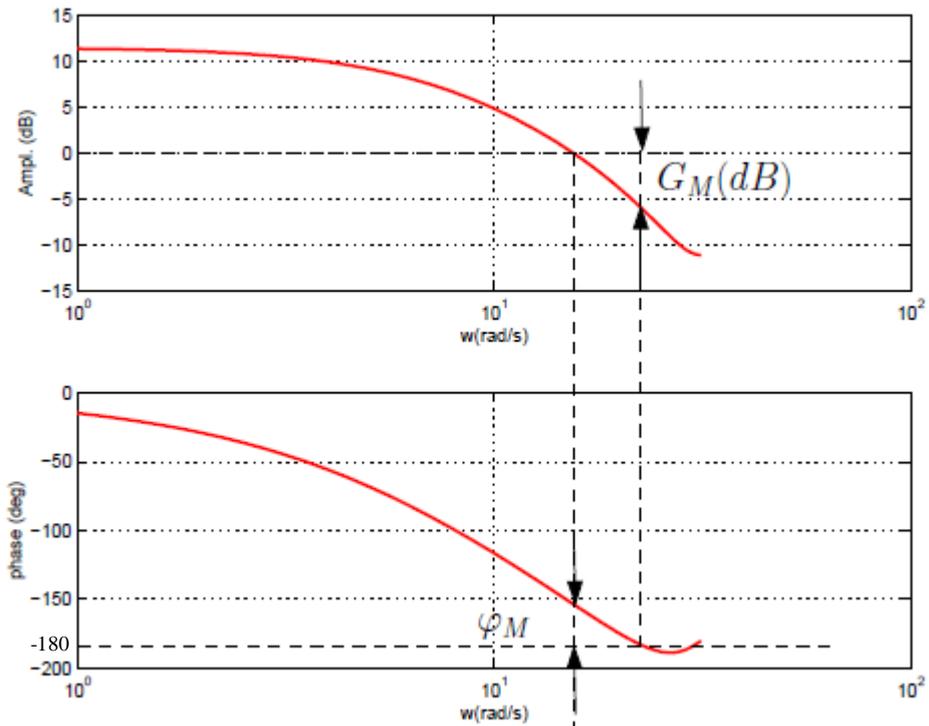


Figure (1) : Figure montrant les marges de gain et de phase sur un diagramme de Bode d'un système discret en boucle ouverte

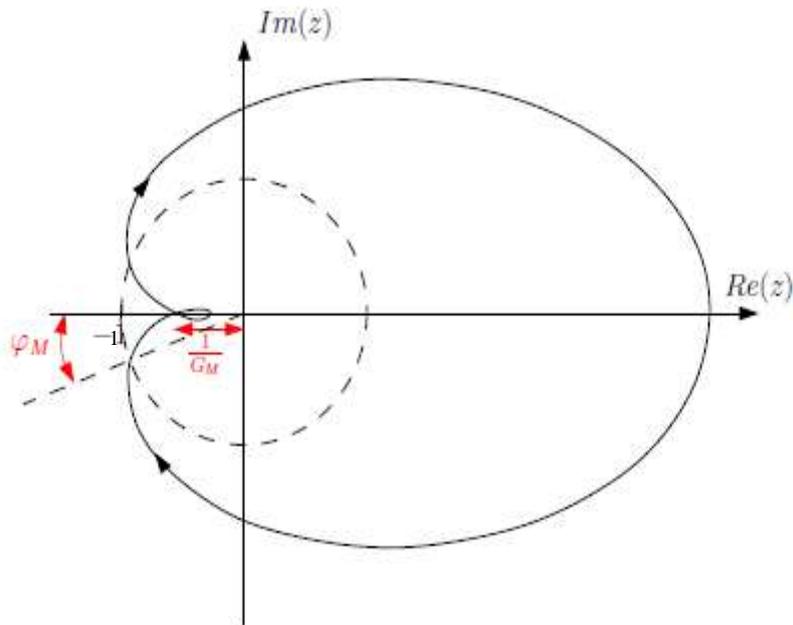


Figure (2) : Figure montrant les marges de gain et de phase sur un lieu de Nyquist d'un système discret en boucle ouverte

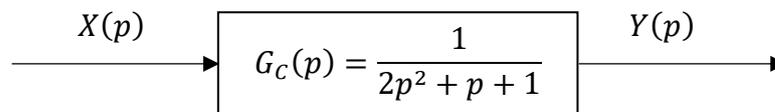
II. Travail demandé

II.1. Préliminaires

1. Lancer MATLAB.
2. Afficher le bureau par défaut de MATLAB.
3. Créer votre dossier personnel correspondant au quatrième TP dans le répertoire C:\Documents\MATLAB portant le nom TPN04.
4. Ajouter votre dossier personnel au Path de MATLAB.

II.2. Application

Soit un système LTI continu du second ordre dont la fonction de transfert est définie comme suit :



Depuis la fenêtre de commande de MATLAB faire les points suivants :

1. Définir la fonction de transfert continue $G_c(p)$.
2. Tracer son diagramme de Bode en boucle ouverte. Puis, déduire sa pulsation de coupure: ' ω_c '.
3. Calculer T_{Min} et T_{Max} caractérisant les bornes de la fourchette définissant la période d'échantillonnage T .
4. En choisissant une période d'échantillonnage convenable ($T_{Min} \leq T \leq T_{Max}$), discrétiser la fonction transfert continue $G_c(p)$ en utilisant un bloqueur d'ordre zéro.
5. Comparer le diagramme de Bode des deux systèmes continu et discret. Commenter.
6. Depuis le diagramme de Bode du système discret, déterminer graphiquement la pulsation de Nyquist. Comparer cette valeur avec celle obtenue analytiquement. Commenter le résultat de comparaison.
7. Depuis le diagramme de Bode du système discret, remplir le tableau suivant :

ω (rad/sec)	0.01	0.5	1	2	3.14
Gain (dB)					
Phase (degré)					

Tableau (1) : Performances fréquentielles obtenues par le diagramme de Bode en boucle ouverte

8. Tracer le lieu de Nyquist du système discret.

9. Depuis le lieu de Nyquist du système discret, remplir le tableau suivant :

w (rad/sec)	0.01	0.5	1	2	3.14
Gain (dB)					
Phase (degré)					

Tableau (2) : Performances fréquentielles obtenues par le lieu de Nyquist en boucle ouverte

10. Comparer les résultats de ces deux tableaux. Commenter.

11. Depuis le diagramme de Bode du système discret, déterminer graphiquement :

- le gain statique ;
- le pic du gain (Peak Response) ;
- la marge de gain et sa pulsation associée (Gain Margin) ;
- la marge de phase et sa pulsation associée (Phase Margin).

12. Comparer les valeurs des marges de gain et des phases obtenues graphiquement avec celles obtenues par l'instruction '*margin*' de la boîte à outils "Control System Toolbox".

13. Depuis le lieu de Nyquist obtenu, déterminer graphiquement :

- le pic du gain (Peak Response) ;
- la marge de gain et sa pulsation associée (Gain Margin) ;
- la marge de phase et sa pulsation associée (Phase Margin).

III. Conclusion

Ce quatrième TP permet d'explorer les diagrammes de Bode et de Nyquist pour évaluer la robustesse des systèmes discrets, en s'appuyant notamment sur les marges de stabilité telles que la marge de gain et la marge de phase. Une comparaison avec les systèmes continus met en évidence les différences de comportement fréquentiel.

TP N°5

Numérisation des contrôleurs PID analogiques réglés par les méthodes de Ziegler-Nichols

Objectif

En utilisant le logiciel MATLAB, savoir faire :

- le réglage des paramètres des contrôleurs P/PI/PID analogiques par les méthodes de Ziegler-Nichols.
- transposition des contrôleurs P/PI/PID analogiques réglés par les méthodes de Ziegler-Nichols en contrôleurs P/PI/PID numériques.

I. Rappels théoriques

I.1. Les contrôleurs PID analogiques

Un contrôleur Proportionnel-Intégral-Dérivé (PID) est un organe de contrôle permettant d'effectuer une régulation en boucle fermée d'un système industriel. Les tableaux suivants montrent les différents types des contrôleurs PID analogiques.

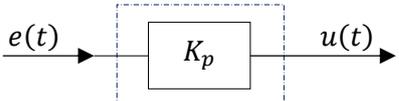
Loi de commande	Fonction de transfert	Schéma fonctionnel
$u(t) = K_p \cdot e(t)$	$C_p(p) = \frac{U(p)}{E(p)} = K_p$	

Tableau (1) : Contrôleur proportionnel P analogique

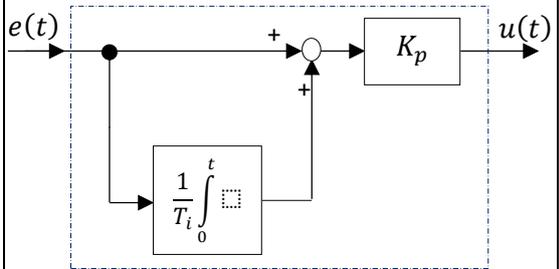
Loi de commande	Fonction de transfert	Schéma fonctionnel
$u(t) = K_p \cdot \left(e(t) + \frac{1}{T_i} \cdot \int_0^t e(\tau) \cdot d\tau \right)$	$C_{PI}(p) = \frac{U(p)}{E(p)} = K_p \cdot \frac{1 + T_i \cdot p}{T_i \cdot p}$	

Tableau (2) : Contrôleur proportionnel-intégral PI analogique mixte

Loi de commande	Schéma fonctionnel
$u(t) = K_p \cdot \left(e(t) + \frac{1}{T_i} \cdot \int_0^t e(\tau) \cdot d\tau + T_d \cdot \frac{de(t)}{dt} \right)$	
Fonction de transfert	
$C_{PID}(p) = K_p \left(1 + \frac{1}{T_i \cdot p} + T_d \cdot p \right)$	

Tableau (3) : Contrôleur proportionnel-intégral-dérivé PID analogique mixte

où K_p est le gain proportionnel. T_i et T_d , sont respectivement, les constantes de temps de l'intégrale et de la dérivée. $e(t)$ et $u(t)$, sont respectivement, l'erreur et la commande.

I.2. Les contrôleurs PID numériques

Les contrôleurs PI/PID numériques peuvent être obtenus par transposition, en utilisant l'approximation d'Euler arrière, de leurs homologues analogiques.

On définit : $E(z)$ et $U(z)$ sont respectivement l'erreur numérique et la commande numérique. T est la période d'échantillonnage.

	Dérivée	Intégrale
Méthode d'Euler arrière	$p \rightarrow \frac{z - 1}{Tz}$	$\frac{1}{p} \rightarrow \frac{Tz}{z - 1}$

Tableau (4) : Approximations de la dérivée et de l'intégrale continues par la méthode d'Euler arrière

Fonction de transfert	Schéma fonctionnel
$C_p(z) = \frac{U(z)}{E(z)} = K_p$	

Tableau (5) : Contrôleur proportionnel P numérique

Fonction de transfert	Schéma fonctionnel
$C_{PI}(z) = \frac{U(z)}{E(z)} = K_p \cdot \left(1 + \frac{T}{T_i} \cdot \frac{z}{z-1} \right)$	

Tableau (6) : Contrôleur proportionnel-intégral PI numérique mixte

Fonction de transfert	Schéma fonctionnel
$C_{PID}(z) = \frac{U(z)}{E(z)} = K_p \left(1 + \frac{T}{T_i} \cdot \frac{z}{z-1} + \frac{T_d}{T} \cdot \frac{z-1}{z} \right)$	

Tableau (7) : Contrôleur proportionnel-intégral-dérivé PID numérique mixte

I.3. Méthodes de réglage de Ziegler-Nichols des contrôleurs PID analogiques

Ziegler et Nichols ont proposé deux approches expérimentales pour ajuster les paramètres des contrôleurs analogiques de type P, PI mixte et PID mixte, l'une dans le domaine temporel où il s'agit d'enregistrer la réponse du système en boucle ouverte (BO), l'autre dans le domaine fréquentiel où il s'agit d'insérer une action proportionnelle de gain K en boucle fermée (BF) et de faire varier ce gain jusqu'à obtenir des oscillations entretenues (d'amener le système à régler en boucle fermée à sa limite de stabilité).

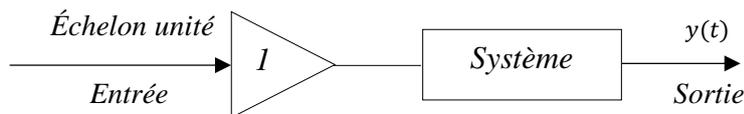


Figure (1) : Essai en boucle ouverte de Ziegler-Nichols

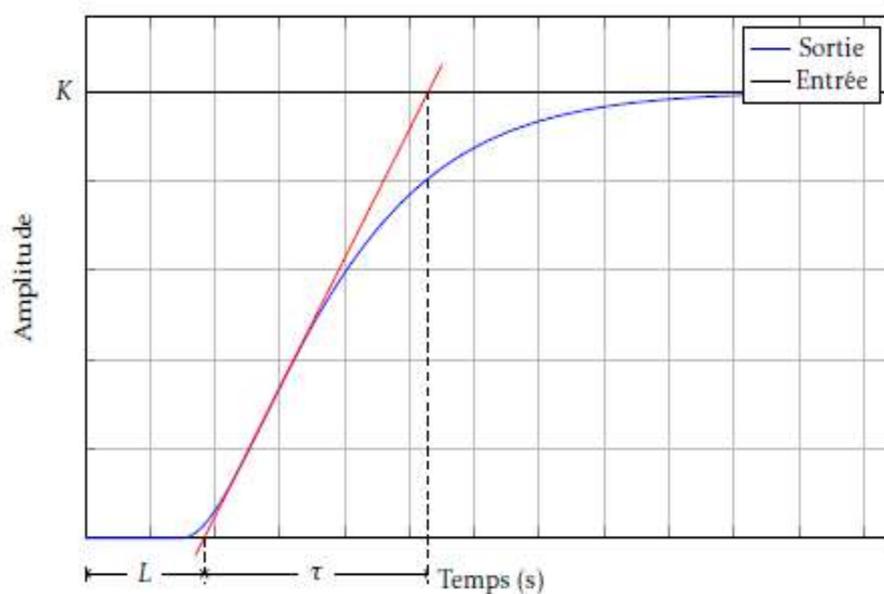


Figure (2) : Réponse indicielle du système en boucle ouverte

Type du contrôleur	Paramètres du contrôleur
Contrôleur P	$K_p = 1/\tau$
Contrôleur PI mixte	$K_p = 0,9/\tau$ $T_i = 3L$
Contrôleur PID mixte	$K_p = 12/\tau$ $T_i = 2L$ $T_d = 0,5L$

Tableau (8) : Approche temporelle : Méthode de la réponse indicielle

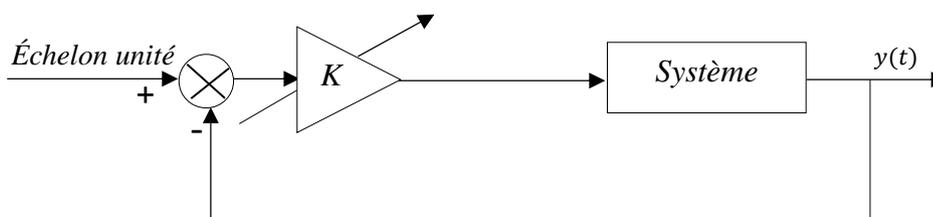


Figure (3) : Essai en boucle fermée de Ziegler-Nichols

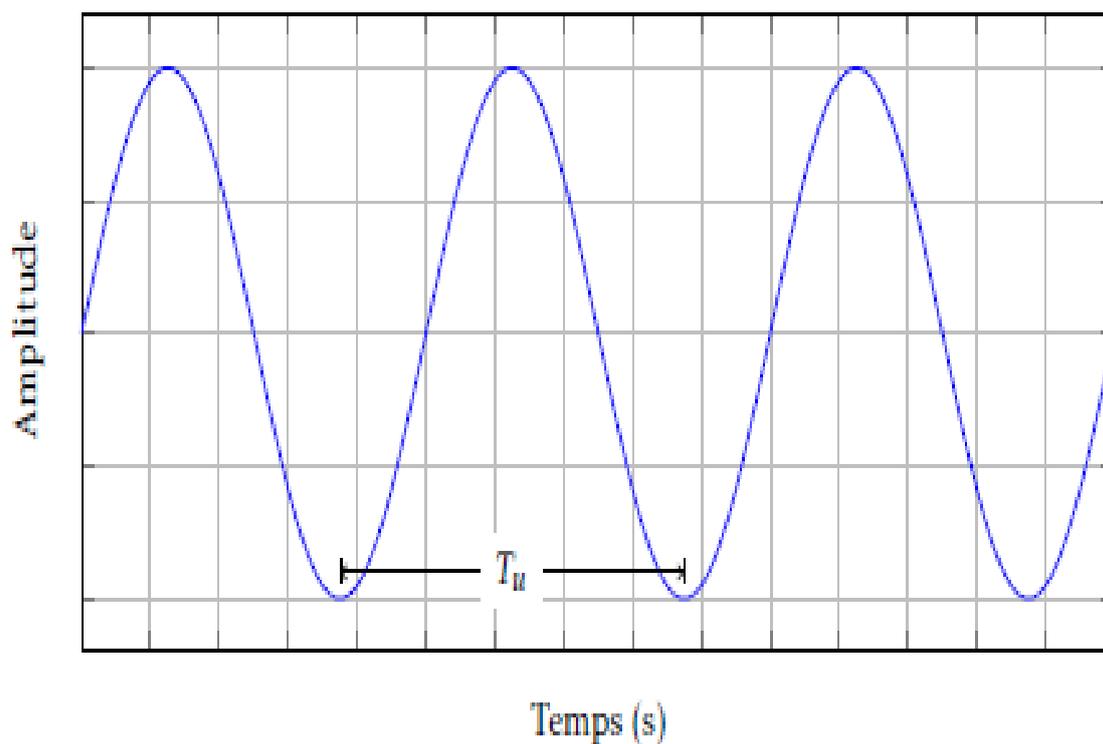


Figure (3) : Régime oscillant du système en boucle fermée

Type du contrôleur	Paramètres du contrôleur
Contrôleur P	$K_p = 0,5K_u$
Contrôleur PI mixte	$K_p = 0,4K_u$ $T_i = 0,8T_u$
Contrôleur PID mixte	$K_p = 0,6K_u$ $T_i = 0,5T_u$ $T_d = 0,125T_u$

Tableau (9) : Approche fréquentielle : Phénomène de pompage

τ et L représentent respectivement la pente de la tangente au point d'inflexion de la réponse indicielle et le retard du système en boucle ouverte (voir la Figure (2)).

K_u et T_u représentent respectivement le gain et la période ultimes du système en boucle fermée (voir la Figure (3)).

II. Travail demandé

II.1. Préliminaires

1. Lancer MATLAB.
2. Afficher le bureau par défaut de MATLAB.
3. Créer votre dossier personnel correspondant au cinquième TP dans le répertoire C:\Documents\MATLAB portant le nom TPN05.
4. Ajouter votre dossier personnel au Path de MATLAB.

II.2. Réglage par l'approche temporelle de Ziegler-Nichols

1. En utilisant un fichier *script*, introduire la fonction de transfert continue suivante: $G(p) = \frac{1}{(1+p)^3}$.
2. Dans le même fichier *script*, tracer la réponse indicielle du système G . Puis, tracer le signal d'échelon unité.
3. Tracer la tangente au point d'inflexion de la réponse indicielle du système.
4. Déterminer graphiquement les paramètres τ et L caractérisant l'approche temporelle de Ziegler-Nichols.
5. En utilisant un autre fichier *script*, calculer par Matlab, les paramètres des contrôleurs de type P, PI et PID (voir le Tableau (8)).
6. En utilisant l'approximation d'Euler arrière, et en prenant comme période d'échantillonnage $T = 1$ (s), numériser les contrôleurs P, PI et PID analogiques calculés par l'approche temporelle de Ziegler-Nichols.
7. Dans le dernier fichier *script*, implémenter les contrôleurs P, PI et PID numériques obtenus.
8. En utilisant les différents contrôleurs analogiques et numériques calculés, simuler la réponse indicielle du système G en boucle fermée.
9. Comparer et interpréter les performances temporelles associées à chaque réponse en remplissant le tableau suivant :

	Premier dépassement	Temps de réponse à 5%	Temps de montée	Erreur statique
Contrôleur P analogique				
Contrôleur P numérique				
Contrôleur PI analogique				
Contrôleur PI numérique				
Contrôleur PID analogique				
Contrôleur PID numérique				

Tableau (10) : Comparaison des performances temporelles obtenues par les différents contrôleurs P, PI et PID analogiques et numériques

II.3. Réglage par l'approche fréquentielle de Ziegler-Nichols

1. Depuis un fichier *script*, en utilisant le tableau (9), calculer les paramètres des contrôleurs P, PI et PID analogiques associés à la méthode fréquentielle de Ziegler-Nichols.
2. Par l'utilisation de l'approximation d'Euler arrière, transposer les contrôleurs P, PI et PID analogiques déjà calculés dans la question 1. On prend comme période d'échantillonnage $T = 1$ (s).
3. Dans le même fichier *script*, implémenter les contrôleurs P, PI et PID numériques obtenus.
4. En utilisant les différents contrôleurs, c.à.d. analogiques et numériques, simuler la réponse indicielle du système G en boucle fermée. Puis, comparer les performances temporelles associées à chaque réponse en remplissant le tableau suivant :

	Premier dépassement	Temps de réponse à 5%	Temps de montée	Erreur statique
Contrôleur P analogique				
Contrôleur P numérique				
Contrôleur PI analogique				
Contrôleur PI numérique				
Contrôleur PID analogique				
Contrôleur PID numérique				

Tableau (11) : Comparaison des performances temporelles obtenues par les différents contrôleurs P, PI et PID analogiques et numériques

III. Conclusion

Ce TP introduit les approches temporelle et fréquentielle de Ziegler-Nichols pour le réglage des contrôleurs PID analogiques, suivies de leur transposition en version numérique. Les simulations réalisées mettent en évidence l'efficacité de ces approches dans l'amélioration des performances des systèmes en boucle fermée.

TP N°6

Réglage des contrôleurs PID numériques par la méthode de Takahashi à l'aide du logiciel MATLAB/Simulink

Objectif

En utilisant le logiciel MATLAB/Simulink, savoir faire le réglage empirique des paramètres des contrôleurs P/PI/PID numériques par la méthode de Takahashi.

I. Rappels théoriques

I.1. Méthode de réglage de Takahashi des contrôleurs PID numériques

Comme Ziegler-Nichols, Takahashi a proposé deux essais, en BO et en BF, pour régler empiriquement les paramètres des contrôleurs P/PI/PID dans le domaine à temps discret.

a. Essai en boucle ouverte 'BO' (Méthode de la réponse indicielle) : cette méthode est basée sur les deux paramètres a et τ de la réponse indicielle issue d'un essai en boucle ouverte.

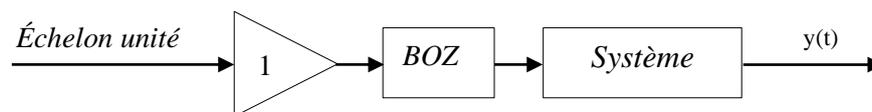


Figure (1) : Essai en boucle ouverte de Takahashi

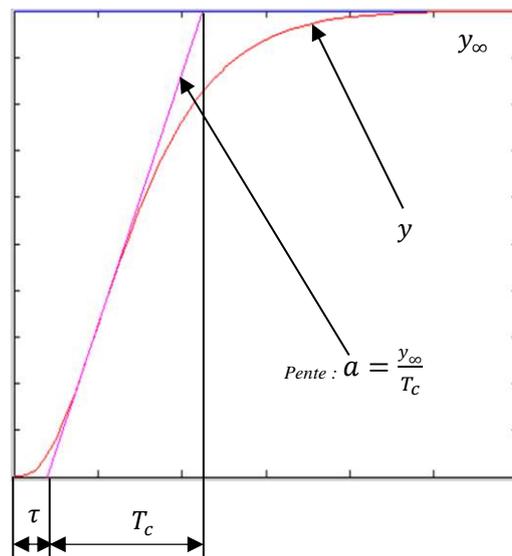


Figure (2) : Réponse indicielle du système en boucle ouverte

Le tableau suivant montre le réglage des contrôleurs P/PI/PID numériques proposé par Takahashi (version en BO) :

Type du contrôleur	Paramètres du contrôleur
Contrôleur P numérique	$K_p = \frac{1}{a(\tau + T)}$
Contrôleur PI numérique	$K_p = \frac{0.9}{a(\tau + 0.5T)} - 0.5K_iT$ $K_i = \frac{0.27}{a(\tau + 0.5T)^2}$
Contrôleur PID numérique	$K_p = \frac{1.2}{a(\tau + 0.5T)} - 0.5K_iT$ $K_i = \frac{0.6}{a(\tau + 0.5T)^2}$ $K_d = \frac{0.5}{a}$

Tableau (1) : Réglage des contrôleurs P/PI/PID numériques par l'essai en boucle ouverte de Takahashi

b. Essai en boucle fermée 'BF' (Phénomène de pompage) : cette méthode est caractérisée par les deux paramètres K_{osc} qui est le gain mettant le système bouclé en oscillation entretenue et T_{osc} qui est la période de l'oscillation entretenue.

Principe : Il s'agit d'augmenter progressivement le gain K d'un correcteur proportionnel pur jusqu'à l'obtention d'une oscillation entretenue.

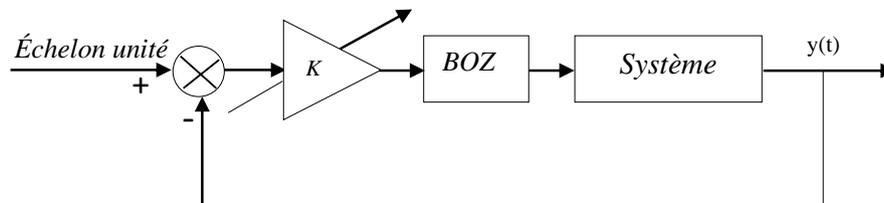


Figure (3) : Essai en boucle fermée de Takahashi

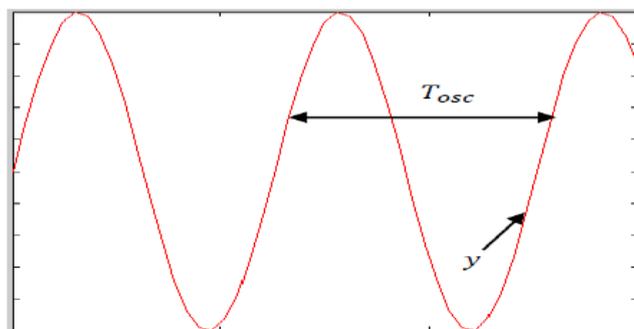


Figure (4) : Régime oscillant du système obtenu en boucle fermée

Le réglage des contrôleurs P/PI/PID numériques proposé par Takahashi (version en BF) est montré par le tableau suivant:

Type du contrôleur	Paramètres du contrôleur
Contrôleur P numérique	$K_p = 0.5K_{osc}$
Contrôleur PI numérique	$K_p = 0.45K_{osc} - 0.5K_iT$ $K_i = 0.54K_{osc}/T_{osc}$
Contrôleur PID numérique	$K_p = 0.6K_{osc} - 0.5K_iT$ $K_i = 1.2K_{osc}/T_{osc}$ $K_d = (3/40)K_{osc}T_{osc}$

Tableau (2) : Réglage des contrôleurs P/PI/PID numériques par l'essai en boucle fermée de Takahashi

où T est la période d'échantillonnage.

I.2. Les contrôleurs PID numériques

Les différents contrôleurs PI et PID numériques que nous allons citer sont obtenus par l'approximation d'Euler arrière et ils ont une structure idéale.

I.2.1. Contrôleur P numérique

La loi de commande d'un contrôleur P numérique est donnée par la relation suivante :

$$U(z) = K_p E(z) \quad (1)$$

I.2.2. Contrôleur PI numérique

Un contrôleur PI numérique peut être décrit suivant une des formes suivantes :

a. Forme pédagogique

$$U(z) = K_p E(z) + K_i T \frac{z}{z-1} E(z) \quad (2)$$

b. Forme industrielle

$$U(z) = K_i T \frac{z}{z-1} E(z) - K_p Y(z) \quad (3)$$

I.2.3. Contrôleur PID numérique

Un contrôleur PID numérique peut être décrit suivant une des formes suivantes :

a. Forme pédagogique

$$U(z) = K_p E(z) + K_i T \frac{z}{z-1} E(z) + \frac{K_d}{T} \frac{z-1}{z} E(z) \quad (4)$$

b. Forme industrielle 1 (PID numérique filtré forme 1)

$$U(z) = K_p E(z) + K_i T \frac{z}{z-1} E(z) - \frac{K_d}{T} \frac{z-1}{z-\alpha} Y(z) \quad (5)$$

c. Forme industrielle 2 (PID numérique filtré forme 2)

$$U(z) = K_i T \frac{z}{z-1} E(z) - \left(K_p + \frac{K_d}{T} \frac{z-1}{z-\alpha} \right) Y(z) \quad (6)$$

Ici $E(z)$, $U(z)$, et $Y(z)$ sont respectivement l'erreur, la commande et la sortie d'un système de contrôle bouclé. K_p , K_i , et K_d représentent respectivement les gains proportionnel, intégral et dérivé. Le choix de la valeur de α est pratiquement de l'ordre de 0.1.

II. Travail demandé

II.1. Préliminaires

1. Lancer MATLAB.
2. Afficher le bureau par défaut de MATLAB.
3. Créer votre dossier personnel correspondant au sixième TP dans le répertoire C:\Documents\MATLAB portant le nom TPN06.
4. Ajouter votre dossier personnel au Path de MATLAB.

II.2. Réglage par l'essai en boucle ouverte (BO) de Takahashi

En utilisant un fichier *script (M-file)* de MATLAB, faire les points suivants:

1. Définir le système LTI continu suivant:

$$G(p) = \frac{1}{p^3 + 3p^2 + 3p + 1}$$

2. Tracer le diagramme de Bode du système G en boucle ouverte.
3. Depuis le diagramme de Bode, déduire la pulsation de coupure ' ω_c ' de $G(p)$.
4. Calculer T_{Min} et T_{Max} caractérisant les bornes de la fourchette définissant la période d'échantillonnage T .

En utilisant Simulink, réaliser le modèle de la figure (5).

5. Afin d'obtenir une réponse aperiodique caractérisant la méthode de l'essai en boucle ouverte de Takahashi, simuler le modèle précédent en choisissant une période d'échantillonnage convenable ($T_{Min} \leq T \leq T_{Max}$).
6. En utilisant un autre fichier *script*, tracer la réponse indicielle du système G discrétisé par un (BOZ) obtenue par le modèle de la figure (5).
7. Dans la même figure, tracer le signal échelon unité.
8. Puis, tracer la tangente au point d'inflexion de la réponse indicielle du système G .
9. Déterminer graphiquement les paramètres a et τ caractérisant l'essai en boucle ouverte de Takahashi.

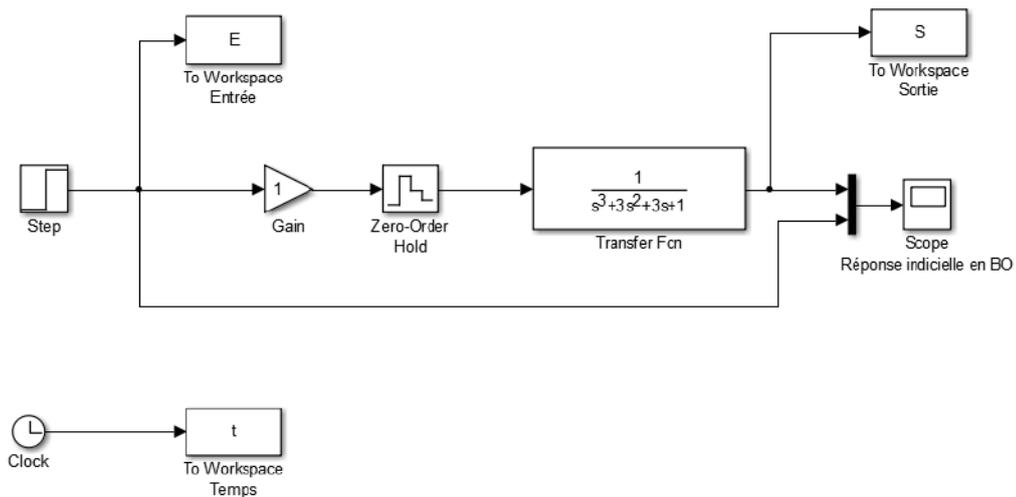


Figure (5) : Modèle Simulink décrivant l'essai en boucle ouverte de Takahashi

10. Calculer par un autre fichier *script*, les paramètres des contrôleurs P/PI/PID numériques proposés par l'essai en boucle ouverte de Takahashi (voir le tableau (1)).

11. Comparer par Simulink, en utilisant les formes pédagogiques des contrôleurs P/PI/PID numériques (c.à.d. les équations (1), (2), et (4)), la réponse indicielle en boucle fermée du système discret. Conclure.

12. Créer un autre modèle Simulink. Puis, comparer la réponse indicielle en boucle fermée du système discret, en utilisant les formes suivantes :

- pour le contrôleur P numérique : équation (1).
- pour le contrôleur PI numérique : équation (3).
- pour le contrôleur PID numérique : équation (5).

Puis, conclure.

13. Dans un autre modèle Simulink, comparer la réponse indicielle en boucle fermée du système discret obtenue par :

- un contrôleur PI numérique régi par l'équation (2).
- un contrôleur PI numérique régi par l'équation (3).

Puis, conclure.

14. Dans un nouveau modèle Simulink, comparer la réponse indicielle en boucle fermée du système discret obtenue par :

- un contrôleur PID numérique décrit par l'équation (4).
- un contrôleur PID numérique décrit par l'équation (5).
- un contrôleur PID numérique décrit par l'équation (6).

Puis, conclure.

II.3. Réglage par l'essai en boucle fermée (BF) de Takahashi

En utilisant Simulink, réaliser le modèle suivant :

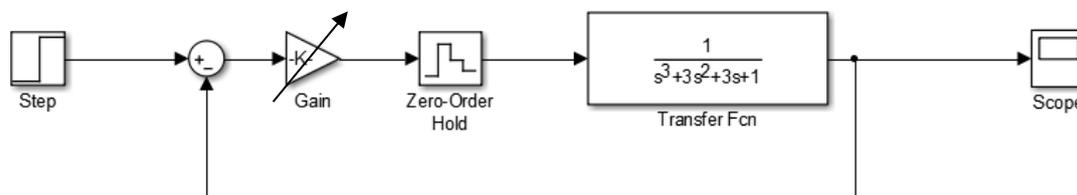


Figure (6) : Modèle Simulink décrivant l'essai en boucle fermée de Takahashi

1. En utilisant la valeur de la période d'échantillonnage choisie dans la section précédente, déterminer les paramètres K_{osc} et T_{osc} caractérisant l'essai en boucle fermée de Takahashi.
2. Depuis l'éditeur de MATLAB en utilisant le tableau (2) de l'essai en boucle fermée de Takahashi, calculer les paramètres des contrôleurs P/PI/PID numériques associés à cette méthode.
3. Créer un autre modèle Simulink. Puis, comparer la réponse indicielle en boucle fermée du système discret, en utilisant les contrôleurs numériques obtenus par l'essai en boucle fermée de Takahashi et conçus par :
 - la forme de l'équation (1) pour le contrôleur P numérique.
 - la forme pédagogique (éq. (2)) pour le contrôleur PI numérique.
 - la forme pédagogique (éq. (4)) pour le contrôleur PID numérique.
4. Créer un nouveau modèle Simulink. Puis, refaire la même comparaison en utilisant :
 - la forme de l'équation (1) pour le contrôleur P numérique.
 - la forme industrielle de l'équation (3) pour le contrôleur PI numérique.
 - la forme industrielle de l'équation (5) pour le contrôleur PID numérique.
5. Dans un autre modèle Simulink, en simulant la réponse indicielle en boucle fermée du système discret, comparer les deux contrôleurs PI numériques régis par les équations (2) et (3). Commenter.
6. Sous un nouveau modèle Simulink, comparer la réponse indicielle en boucle fermée du système discret, en utilisant les trois formes des contrôleurs PID numériques. Puis, conclure.

II.4. Essai en BO de Takahashi VS essai en BF de Takahashi

II.4.1. Comparaison des contrôleurs P numériques

Sous un nouveau modèle Simulink, simuler et comparer la réponse indicielle en boucle fermée, du système discret, obtenue par les contrôleurs :

- P numérique obtenu par l'essai en boucle ouverte de Takahashi.
- P numérique obtenu par l'essai en boucle fermée de Takahashi.

II.4.2. Comparaison des contrôleurs PI numériques

En utilisant un deuxième modèle Simulink simuler et comparer la réponse indicielle en boucle fermée, du système discret, obtenue par les différents contrôleurs PI numériques à savoir :

- le PI décrit par la forme pédagogique et obtenu par l'essai en boucle ouverte de Takahashi.
- le PI décrit par la forme industrielle et obtenu par l'essai en boucle ouverte de Takahashi.
- le PI décrit par la forme pédagogique et obtenu par l'essai en boucle fermée de Takahashi.
- le PI décrit par la forme industrielle et obtenu par l'essai en boucle fermée de Takahashi.

II.4.3. Comparaison des contrôleurs PID numériques

En simulant sous Simulink la réponse indicielle en boucle fermée du système discret, on veut comparer les contrôleurs de type PID numérique obtenus par les deux essais de Takahashi et conçus par les différentes formes. Pour ce faire, interpréter les différents résultats obtenus, puis tirer par comparaison :

- le meilleur contrôleur ;
- la meilleure forme.

III. Conclusion

La méthode de Takahashi a été appliquée pour le réglage empirique de contrôleurs PID numériques sur un système du troisième ordre. Les essais en boucle ouverte et en boucle fermée ont permis de comparer différentes structures de régulateurs (pédagogique et industrielle) et d'identifier celle offrant les performances les plus adaptées au comportement dynamique du système.

TP N°7

Régulation de la vitesse d'un moteur à courant continu par des contrôleurs PID numériques

Objectif

L'objectif de ce TP est de mettre en place des correcteurs PID numériques par placement de pôles pour régler la vitesse d'un moteur à courant continu (MCC).

I. Rappels théoriques

I.1. Modélisation d'un moteur à courant continu (MCC)

On considère un moteur à courant continu à aimant permanent commandé par l'induit dont le schéma électrique et mécanique équivalent de l'induit est présenté par la figure suivante:

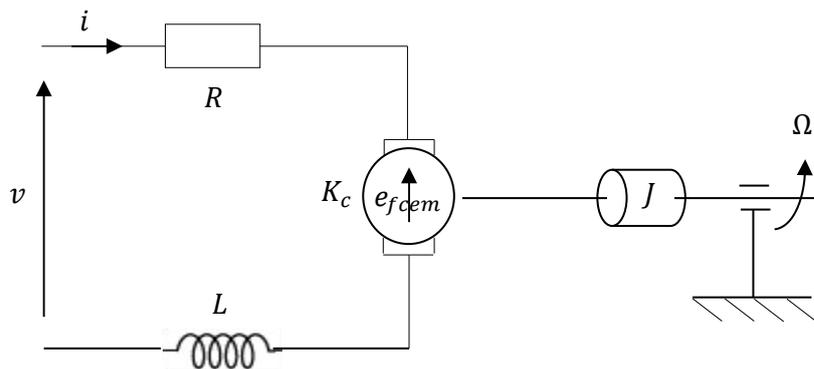


Figure (1) : Schéma électrique et mécanique de l'induit

Les équations régissant le fonctionnement du moteur à courant continu à aimant permanent sont les suivantes :

- Équation de la partie électrique de l'induit

$$v(t) = Ri(t) + L \frac{di(t)}{dt} + e_{fcem}(t) \quad (1)$$

- Équation de la force contre-électromotrice

$$e_{fcem}(t) = K_e \Omega(t) \quad (2)$$

- Équation du couple électromagnétique

$$C_m(t) = K_c i(t) \quad (3)$$

- Équation de la partie mécanique de l'induit

$$C_m(t) = J \frac{d\Omega(t)}{dt} + C_r(t) \quad (4)$$

Ces équations peuvent être regroupées dans le système suivant :

$$\begin{cases} v(t) = Ri(t) + L \frac{di(t)}{dt} + K_e \Omega(t) \\ C_m(t) = K_c i(t) \\ J \frac{d\Omega(t)}{dt} = C_m(t) - C_r(t) \\ C_r(t) = R_f \Omega(t) \end{cases} \quad (5)$$

En appliquant l'opérateur de Laplace sur (5), avec conditions initiales nulles, on obtient le système suivant :

$$\begin{cases} V(p) = RI(p) + LpI(p) + K_e \Omega(p) \\ C_m(p) = K_c I(p) \\ Jp\Omega(p) = C_m(p) - R_f \Omega(p) \\ C_r(p) = R_f \Omega(p) \end{cases} \quad (6)$$

Deux modèles peuvent être envisagés pour le moteur :

- Modèle du premier ordre

En négligeant l'effet du frottement visqueux et l'inductance de la bobine de l'induit, le système (6) est décrit alors par le schéma fonctionnel suivant :

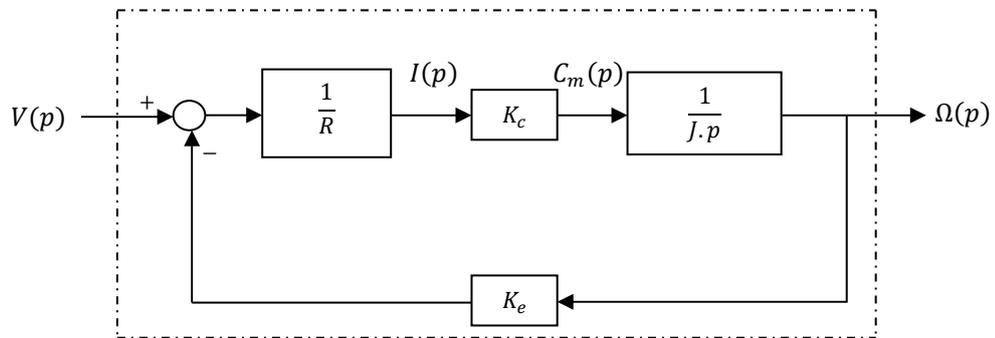


Figure (2) : Schéma fonctionnel d'un moteur MCC décrit par un premier ordre

Depuis ce schéma fonctionnel, on peut tirer facilement la fonction de transfert liant la vitesse du moteur et la tension de l'induit :

$$\frac{\Omega(p)}{V(p)} = \frac{K}{1+\tau p} \quad (7)$$

où :

$K = \frac{1}{K_e}$ est le gain statique du moteur et $\tau = \frac{R.J}{K_e.K_c}$ est la constante du temps du moteur.

- Modèle du second ordre

Si on néglige juste l'effet du frottement visqueux dans ce cas le moteur est représenté par le schéma fonctionnel suivant :

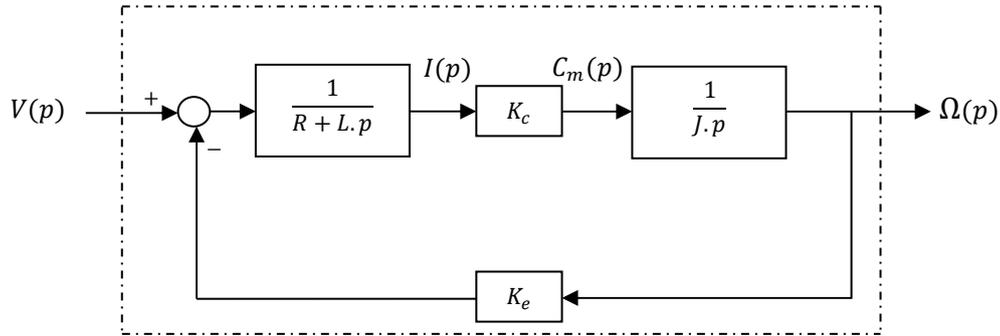


Figure (3) : Schéma fonctionnel d'un moteur MCC décrit par un second ordre

Dans ce cas, la vitesse du moteur par rapport à la tension de l'induit est décrite par une fonction de transfert du second ordre comme suit :

$$\frac{\Omega(p)}{V(p)} = \frac{K}{1 + \frac{2\xi}{\omega_0}p + \frac{1}{\omega_0^2}p^2} \quad (8)$$

avec :

$K = \frac{1}{K_e}$ est le gain statique du moteur, $\xi = \frac{R}{2} \sqrt{\frac{J}{K_e \cdot K_c \cdot L}}$ est le facteur d'amortissement du moteur et $\omega_0 = \sqrt{\frac{K_e \cdot K_c}{L \cdot J}}$ est la pulsation propre du moteur.

Les paramètres et les variables du moteur sont définis comme suit : Résistance de l'induit: $R = 1.84 \text{ Ohm}$, Inductance de l'induit: $L = 0.00077 \text{ H}$, Constante de couple: $K_c = 0.9$, Constante de f_{cem} : $K_e = 0.9$, Moment d'inertie: $J = 0.0061 \text{ kg.m}^2$, $v(t)$ est la tension de l'induit, $i(t)$ est le courant de l'induit, $\Omega(t)$ est la vitesse angulaire du moteur, f_{cem} est la force contre-électromotrice, $C_m(t)$ est le couple électromagnétique, $C_r(t)$ est le couple de charge et R_f est le coefficient de frottement visqueux.

I.2. Synthèse des contrôleurs PID numériques par placement de pôles

I.2.1. Commande d'un premier ordre discret par un contrôleur PI numérique

Considérons un système discret du premier ordre commandé par un contrôleur PI numérique, comme le montre le schéma-bloc suivant :

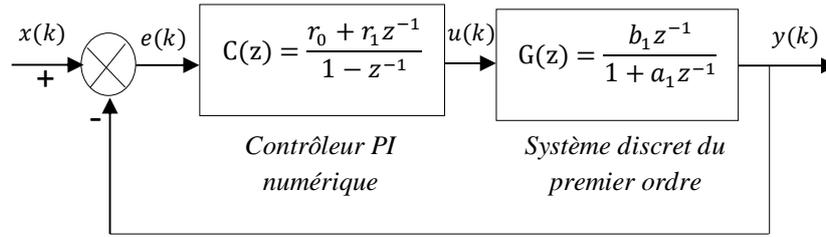


Figure (4) : Schéma-bloc d'une commande d'un premier ordre discret par un contrôleur PI numérique

En utilisant le principe de la méthode de placement de pôles, deux comportements en boucle fermée peuvent être envisagés : comportement en premier ordre et comportement en second ordre.

- Comportement en premier ordre

L'idée de cette méthode consiste à compenser le pôle du système du premier ordre par le zéro du contrôleur PI numérique. Ce qui amène un comportement en boucle fermée du premier ordre.

Dans ce cas, en utilisant le principe de placement de pôles, le réglage des paramètres du contrôleur PI numérique est donné comme suit :

$$r_0 = \frac{1-\lambda_0}{b_1} \quad (9)$$

$$r_1 = a_1 r_0 \quad (10)$$

avec $\lambda_0 = e^{-\frac{T}{\tau}}$. T et τ représentent respectivement la période d'échantillonnage et la constante du temps du système continu du premier ordre imposé.

- Comportement en second ordre

Ici, le principe consiste à obtenir en boucle fermée un système assimilable à un second ordre continu. En utilisant le principe de placement de pôles, les coefficients du contrôleur PI numérique sont donnés par les relations suivantes :

$$r_0 = \frac{1-a_1+\rho_1}{b_1} \quad (11)$$

$$r_1 = \frac{1+\rho_2}{b_1} \quad (12)$$

Les valeurs de ρ_1 et ρ_2 sont calculées comme suit :

Si la dynamique désirée en boucle fermée est apériodique, il faut choisir deux constantes de temps : τ_1 et τ_2 , telles que :

$$\rho_1 = -(\lambda_1 + \lambda_2), \quad \rho_2 = \lambda_1 \lambda_2, \quad \lambda_1 = e^{-\frac{T}{\tau_1}} \quad \text{et} \quad \lambda_2 = e^{-\frac{T}{\tau_2}} \quad (13)$$

Si la dynamique désirée est oscillatoire, il faut choisir une pulsation propre ω_0 et un coefficient d'amortissement ξ :

$$\rho_1 = -2e^{-rT} \cos(cT), \quad \rho_2 = e^{-2rT}, \quad r = \xi \omega_0 \text{ et } c = \omega_0 \sqrt{1 - \xi^2} \quad (14)$$

I.2.2. Commande d'un second ordre discret par un contrôleur PID numérique filtré

La figure (5) montre une commande d'un système discret du second ordre par un contrôleur PID numérique filtré.

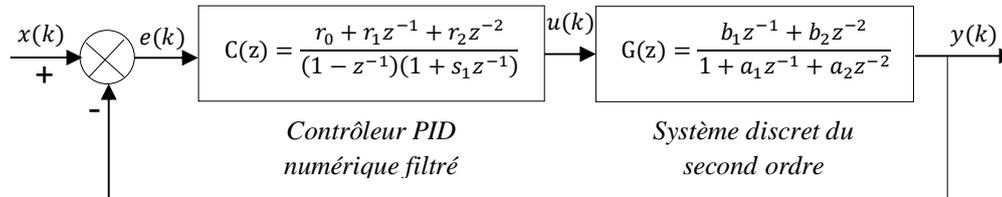


Figure (5) : Schéma-bloc d'une commande d'un second ordre discret par un contrôleur PID numérique filtré

En choisissant un comportement du second ordre en boucle fermée, les paramètres du contrôleur PID numérique filtré sont calculés par le principe de placement de pôles:

$$r_0 = \frac{1 + \rho_1 + \rho_2}{b_1 + b_2}, \quad r_1 = a_1 r_0, \quad r_2 = a_2 r_0 \text{ et } s_1 = b_2 r_0 - \rho_2 \quad (15)$$

II. Travail demandé

II.1. Préliminaires

1. Lancer MATLAB.
2. Afficher le bureau par défaut de MATLAB.
3. Créer votre dossier personnel correspondant au septième TP dans le répertoire C:\Documents\MATLAB portant le nom TPN07.
4. Ajouter votre dossier personnel au Path de MATLAB.

II.2. Commande PI numérique de la vitesse du moteur décrit par son modèle du premier ordre

Dans cette première section de TP, en se basant sur le schéma-bloc de la figure (4) nous voulons commander le moteur, décrit par la version discrète de son modèle du premier ordre selon l'équation (7).

1. Créer un fichier *script* (*M-file*) de MATLAB contenant les paramètres du moteur. Sauvegarder le *script* sous le nom "ParametresMCC".
2. Dans un autre fichier *script*, définir la fonction de transfert continu du premier ordre entre la vitesse et la tension du moteur MCC (équation (7)).
3. Tracer son diagramme de Bode en boucle ouverte.
4. D'après le diagramme de Bode, déduire sa pulsation de coupure ' ω_c '.

5. Calculer T_{Min} et T_{Max} caractérisant les bornes de la fourchette définissant la période d'échantillonnage T .

6. En choisissant une période d'échantillonnage convenable ($T_{Min} \leq T \leq T_{Max}$), discrétiser la fonction du moteur vitesse/tension en utilisant un bloqueur d'ordre zéro.

7. On désire commander la vitesse du moteur par un contrôleur PI numérique, pour ce faire nous allons choisir tout d'abord un comportement en premier ordre quatre fois plus rapide que le système continu décrivant le moteur (vitesse/tension).

a. Créer un deuxième fichier *script* (*M-file*) portant le nom “*VitesseMCC_PIN1*”.

b. Dans ce fichier, calculer le contrôleur PI numérique correspondant (PIN1).

c. En utilisant un échelon d'amplitude de 20 (V), simuler la réponse de la vitesse du moteur discrétisé commandé par le contrôleur ‘PIN1’. Puis dans la même figure tracer le signal d'entrée.

d. Évaluer la stabilité du système discrétisé en boucle ouverte et en boucle fermée. Puis commenter.

8. Maintenant, nous allons considérer que le comportement en boucle fermée de la commande de la vitesse du moteur par un contrôleur PI numérique est assimilable à un deuxième ordre discret.

8.1. En premier lieu, nous souhaitons avoir une dynamique du second ordre apériodique, pour ce faire :

a. Créer un nouveau fichier *script* portant le nom “*VitesseMCC_PIN2*”.

b. En choisissant deux constantes de temps τ_1 et τ_2 , calculer le contrôleur PI numérique correspondant (PIN2).

c. Pour un échelon d'amplitude 20 (V) simuler la réponse de la vitesse du moteur discrétisé commandé par le contrôleur ‘PIN2’. Tracer dans la même figure le signal d'entrée.

d. Évaluer la stabilité du système discrétisé en boucle ouverte et en boucle fermée. Puis conclure.

8.2. En deuxième lieu, on veut réaliser une dynamique oscillatoire en choisissant une pulsation propre $\omega_0 = 25$ (rad/s) et un coefficient d'amortissement $\xi = \sqrt{2}/2$.

a. Créer un autre fichier *script* portant le nom “*VitesseMCC_PIN3*”.

b. Dans ce fichier implémenter le contrôleur PI numérique correspondant (PIN3).

c. En utilisant le même échelon, simuler la réponse de la vitesse du moteur discrétisé commandé par ce nouveau contrôleur ‘PIN3’.

d. Évaluer la stabilité du système discrétisé en boucle ouverte et en boucle fermée. Puis commenter.

8.3. Dans un autre fichier MATLAB simuler et comparer les réponses de la vitesse du moteur obtenues par les différents contrôleurs PI numériques calculés. Puis, comparer les performances temporelles associées à chaque réponse en remplissant le tableau suivant :

	<i>Premier dépassement</i>	<i>Temps de réponse à 5%</i>	<i>Temps de montée</i>	<i>Erreur statique</i>
<i>Contrôleur 'PIN1'</i>				
<i>Contrôleur 'PIN2'</i>				
<i>Contrôleur 'PIN3'</i>				

Tableau (1) : Comparaison des performances temporelles obtenues par les différents contrôleurs

II.3. Commande PID numérique filtré de la vitesse du moteur décrit par son modèle du second ordre

Dans cette section de TP, nous voulons commander le moteur par un contrôleur PID numérique filtré en se basant sur le schéma-bloc de la figure (5) en utilisant la version discrétisée du modèle du second ordre de l'équation (8). Dans ce cas, on impose un comportement du second ordre en boucle fermée. Par conséquent et comme la section précédente, nous choisissons la pulsation propre $\omega_0 = 25$ (rad/s) et le coefficient d'amortissement $\xi = \sqrt{2}/2$.

1. Dans un autre fichier *script* définir la fonction de transfert continu du second ordre entre la vitesse et la tension du moteur MCC donnée par l'équation (8).
2. Tracer son diagramme de Bode en boucle ouverte.
3. Puis, d'après le diagramme de Bode déduire sa pulsation de coupure ' ω_c '. Puis, calculer T_{Min} et T_{Max} caractérisant les bornes de la fourchette définissant la période d'échantillonnage T .
4. En choisissant une période d'échantillonnage convenable ($T_{Min} \leq T \leq T_{Max}$), discrétiser la fonction du moteur vitesse/tension en utilisant un BOZ.
5. Implémenter un contrôleur PID numérique filtré en se basant sur les relations citées dans (15). Puis, simuler la réponse de la vitesse du moteur en utilisant un échelon d'amplitude 20 (V).
6. Évaluer la stabilité du système discret en boucles ouverte et fermée. Puis, commenter.

III. Conclusion

Ce TP a porté sur la modélisation d'un moteur à courant continu (MCC) ainsi que sur la synthèse de plusieurs contrôleurs numériques, comprenant trois contrôleurs PI et un contrôleur PID filtré. Ces contrôleurs ont été conçus par placement de pôles afin d'assurer une régulation précise et stable de la vitesse du moteur MCC.

TP N°8
Commande PID numérique sur structure polynomiale RST

Objectif

Savoir mettre en place des contrôleurs PID numériques filtrés modélisés sous forme polynomiale RST.

I. Rappels théoriques sur les contrôleurs PID numériques filtrés modélisés sous forme polynomiale RST

On considère le schéma d'une commande par un contrôleur PID numérique filtré suivant :

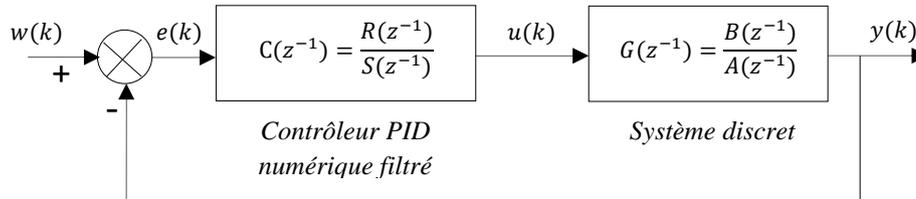


Figure (1) : Schéma d'une commande d'un système discret par un PID numérique filtré

Dans ce schéma le contrôleur PID numérique filtré prend la forme filtrée suivante :

$$C(z^{-1}) = \frac{R(z^{-1})}{S(z^{-1})} \quad (1)$$

où

$$R(z^{-1}) = r_0 + r_1 z^{-1} + r_2 z^{-2} \quad (2)$$

$$S(z^{-1}) = (1 - z^{-1})(1 + s_1 z^{-1}) \quad (3)$$

Ce contrôleur possède quatre paramètres réglables, qui sont : r_0 , r_1 , r_2 et s_1 . En fait, ces paramètres peuvent être réglés en utilisant la stratégie de placement de pôles.

La fonction de transfert en boucle fermée issue du schéma de la figure (1) prend la forme :

$$\begin{aligned} F(z) &= \frac{Y(z)}{W(z)} = \frac{B(z^{-1}) R(z^{-1})}{A(z^{-1}) S(z^{-1}) + B(z^{-1}) R(z^{-1})} \\ &= \frac{B(z^{-1}) R(z^{-1})}{P(z^{-1})} \end{aligned} \quad (4)$$

avec $P(z^{-1})$: polynôme imposé en fonction des spécifications désirées (temps de montée, amortissement, dépassement maximale, erreur statique, etc...).

D'autre part, ce contrôleur PID numérique filtré peut être modélisé sous forme d'un contrôleur polynomial RST comme suit :

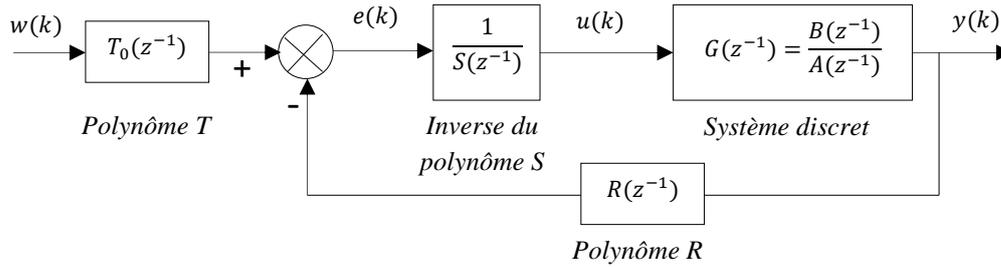


Figure (2) : Commande PID numérique sur structure polynomiale RST

où $R(z^{-1})$, $S(z^{-1})$, et $T_0(z^{-1})$ sont les trois polynômes en ' z^{-1} ' constituant la structure polynomiale RST.

Dans ce cas, la fonction de transfert du système bouclé (Figure (2)) est calculée comme suit:

$$F(z) = \frac{Y(z)}{W(z)} = \frac{T_0(z^{-1}) B(z^{-1})}{A(z^{-1}) S(z^{-1}) + B(z^{-1}) R(z^{-1})} \quad (5)$$

Afin de déterminer les deux polynômes $R(z^{-1})$ et $S(z^{-1})$, il suffit de résoudre l'équation de Bezout (6), en imposant un polynôme de référence $P(z^{-1})$.

$$P(z^{-1}) = A(z^{-1}) S(z^{-1}) + B(z^{-1}) R(z^{-1}) \quad (6)$$

En effet, si on veut imposer un gain statique unité à la fonction de transfert (5) en prenant :

$$T_0(z^{-1}) = \frac{P(1)}{B(1)} \quad (7)$$

Et comme $S(1) = 0$, il vient $P(1) = B(1)$ par conséquent, l'équation (7) devient alors:

$$T_0(z^{-1}) = R(1) = r_0 + r_1 + r_2 \quad (8)$$

II. Travail demandé

II.1. Préliminaires

1. Lancer MATLAB.
2. Afficher le bureau par défaut de MATLAB.
3. Créer votre dossier personnel correspondant au huitième TP dans le répertoire C:\Documents\MATLAB portant le nom TPN08.
4. Ajouter votre dossier personnel au Path de MATLAB.

II.2. Applications

II.2.1. Application 1 : Étude en boucle ouverte

Soit un système LTI continu du second ordre dont la fonction de transfert est définie comme suit :

$$G_c(p) = \frac{1}{0.1p^2 + p}$$

En utilisant un fichier *script*, faire les points suivants :

1. Définir la fonction de transfert continue $G_c(p)$.
2. Trouver la pulsation de coupure ' ω_c du système $G_c(p)$ depuis son diagramme de gain de Bode en boucle ouverte.
3. En choisissant une période d'échantillonnage convenable T dans la fourchette $[T_{Min}, T_{Max}]$, discrétiser la fonction transfert continue $G_c(p)$ en utilisant un bloqueur d'ordre zéro.
4. Réécrire le système discret obtenu en fonction de puissances ' z^{-1} '.
5. Évaluer la stabilité en boucle ouverte du système discret obtenu.
6. Puis, identifier ses pôles/zéros stables et instables. Conclure sur la stabilité en boucle ouverte.

II.2.2. Application 2 : Étude en boucle fermée

II.2.2.1. Commande par un contrôleur PID numérique filtré

D'abord, on veut commander le système discret obtenu dans la question 4 de la section précédente en se basant sur le schéma de la figure (1) par un contrôleur PID numérique filtré. On désire que le système bouclé obtenu se comporte comme un second ordre associé avec un facteur d'amortissement $\xi = 0.707$ et une pulsation propre $\omega_0 = 0.6 \text{ rad/s}$.

Dans le programme MATLAB précédent faire:

1. Introduire la fonction de transfert imposée en continu.
2. En utilisant la même période d'échantillonnage choisie dans la section précédente, discrétiser cette dernière fonction par un bloqueur d'ordre zéro.
3. Calculer le polynôme caractéristique du système discret imposé : $P(z^{-1})$.
4. Écrire l'équation (6) sous forme matricielle : $M \cdot \phi = P$ de telles sortes que le vecteur ϕ contient les paramètres du contrôleur PID numérique filtré. Puis, résoudre le système matriciel obtenu en tirant les valeurs des paramètres r_0, r_1, r_2 et s_1 .
5. Calculer les deux polynômes $R(z^{-1})$ et $S(z^{-1})$ en utilisant les équations (2) et (3).
6. Calculer la fonction de transfert du contrôleur PID numérique filtré en utilisant l'équation (1).
7. Le contrôleur PID numérique filtré obtenu, est-il physiquement réalisable en temps réel ? Est-il stable ? Justifier vos réponses.
8. En utilisant le schéma de la figure (1), simuler la réponse indicielle du système discret.
9. Évaluer la stabilité du système bouclé.

II.2.2.2. Commande par un contrôleur PID numérique filtré modélisé sous forme RST

Dans le but de commander les systèmes discrets par des contrôleurs polynomiaux de type RST, nous allons modéliser le contrôleur PID numérique filtré obtenu dans la section précédente sous forme d'un contrôleur polynomial RST. En se basant sur le schéma de la figure (2) et en conservant le même comportement imposé précédemment en boucle fermée, compléter le programme MATLAB précédent :

1. En utilisant l'équation (8), calculer le polynôme $T_0(z^{-1})$.
2. En utilisant le schéma de la figure (2), implémenter les différents polynômes. Puis, simuler la réponse indicielle du système discret.
3. Évaluer la stabilité du système bouclé obtenu dans ce cas de figure.

II.2.2.3. Comparaison entre le PID numérique filtré et le PID modélisé sous forme RST

Toujours dans le même programme faire les points suivants:

1. Comparer la réponse indicielle obtenue par les deux contrôleurs :
 - le PID numérique filtré ;
 - le PID numérique filtré modélisé sous forme RST.
2. Déterminer graphiquement :
 - le premier dépassement (Peak Response) ;
 - le temps de réponse à 5% (Settling Time) ;
 - le temps de montée (Rise Time) ;
 - l'erreur statique (Steady State error).
3. Puis remplir le tableau ci-dessous et comparer les différentes performances temporelles mesurées.
4. Déduire sur la stabilité obtenue par ces contrôleurs. Puis, déduire le meilleur contrôleur parmi les deux qu'on a développé.

	<i>Premier dépassement</i>	<i>Temps de réponse</i>	<i>Temps de montée</i>	<i>Erreur statique</i>
<i>Contrôleur PID numérique filtré</i>				
<i>Contrôleur PID numérique filtré modélisé sous forme RST</i>				

Tableau (1) : Comparaison des performances temporelles obtenues par les deux contrôleurs calculés

III. Conclusion

Ce dernier TP porte sur la modélisation de contrôleurs PID numériques filtrés selon une structure polynomiale RST numérique. L'équation de Bézout est utilisée pour imposer un comportement dynamique souhaité en boucle fermée, avec une comparaison des performances entre les régulateurs RST numériques et les PID numériques filtrés.

Conclusion générale

Ce polycopié adopte une démarche progressive et rigoureuse visant à introduire et consolider les concepts fondamentaux de la régulation numérique à travers une série de travaux pratiques basés sur le logiciel MATLAB/Simulink. Il offre un cadre structuré pour l'exploration approfondie d'un ensemble cohérent de notions, incluant la modélisation, l'analyse temporelle et fréquentielle, l'évaluation de la stabilité des systèmes LTI discrets, ainsi que la synthèse de contrôleurs analogiques et numériques de type P, PI, PID, en complément de structures de commande avancées telles que les régulateurs polynomiaux RST. L'ensemble de ces activités permettra aux étudiants de mobiliser leurs acquis théoriques afin d'analyser et piloter efficacement des systèmes industriels.

Références

- [1]: J. M. Retif, ‘Synthèse d’une commande robuste : Correcteurs échantillonnés, commande par P.I.D., par modèle d’état et polynomiale RST-Automatique’, *Ellipses Marketing*, 2011.
- [2]: Y. Takahashi, C. Chan, D. Auslander, ‘Parametereinstellung bei linearen DDCAgorithme’, *Regelungstech. Prozessdatenver.*, 19, 237-284, 1971.
- [3]: P. Borne, G.D. Tanguv, J. P. Richard, F. Rotella, I. Zambetalcis, ‘Analyse et régulation de processus industriels - Régulation numérique’, *Tome 2, Editions Technip*, 1993.
- [4]: Y. Sévely, ‘Systèmes et Asservissements Linéaires Echantillonnés’, 2^e Edition, *Dunod, Paris*, 1973.
- [5]: L. Kharroubi, ‘Éléments de Régulation Numérique, Cours et Exercices’, *Polycopie, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf (USTO-MB)*, 2018.
- [6]: L. Kharroubi, ‘Contribution à la Modélisation et à la Commande Intelligente d’un Système Chariot-Pendule’, *Thèse, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf (USTO-MB)*, 2017.
- [7]: J. J. Di Stefano, A. R. Stubberrud, I. J. Williams, ‘SYSTEMES ASSERVIS 1, Cours et Problèmes’, *SERIE SCHAUM, Engineering Department, University of California, Los Angeles*, 1983.
- [8]: J. J. Di Stefano, A. R. Stubberrud, I. J. Williams, ‘SYSTEMES ASSERVIS 2, Cours et Problèmes’, *SERIE SCHAUM, Engineering Department, University of California, Los Angeles*, 1983.
- [9]: Gonzalo Cabodevila, ‘Automatique linéaire échantillonnée’, 2009.
http://jmfriedt.free.fr/Gonzalo_cours1A.pdf.
- [10]: Y. B. Ensica, ‘Commande des systèmes par calculateur’,
<https://personnel.isae-supaero.fr/IMG/pdf/LectureNotes.pdf>.
- [11]: G. Cormier ‘Controlleurs’, Chapitre,
https://www.academia.edu/38140708/GELE5313_Notes

1. Principales commandes de la boîte à outils "Control System Toolbox"

1.1. Définition et association des systèmes linéaires (LTI)

tf	- crée un système sous une représentation fonction de transfert polynomiale.
s=tf('s')	- indique à MATLAB que s est la variable de Laplace.
p=tf('p')	- indique à MATLAB que p est la variable de Laplace.
z=tf('z')	- indique à MATLAB que z est la variable complexe.
z=tf('z',T)	- indique à MATLAB que z est la variable complexe et T est la période d'échantillonnage.
zpk	- crée ou convertit un système sous une représentation zéro-pôle et gain.
ss	- crée ou convertit un système sous une représentation d'état.
ord2	- crée un système continu d'ordre 2.
tfddata	- extrait le numérateur et le dénominateur d'une fonction de transfert.
zpkdata	- calcul des zéros, des pôles et du gain d'une fonction de transfert.
ssdata	- renvoie les matrices A, B, C et D du modèle d'état d'un système.
series	- connecte des systèmes en série.
parallel	- connecte des systèmes en parallèle.
feedback	- connecte des systèmes en boucle fermée.
inv	- inverse une fonction de transfert.

1.2. Réponses temporelles des systèmes LTI

step	- réponse indicielle d'un système continu ou discret.
dstep	- réponse indicielle d'un système discret.
impulse	- réponse impulsionnelle d'un système continu ou discret.
dimpulse	- réponse impulsionnelle d'un système discret.
lsim	- réponse à une entrée quelconque d'un système continu.
dlsim	- réponse à une entrée quelconque d'un système discret.
initial	- condition initiale pour une réponse temporelle d'un système continu.
dinitial	- condition initiale pour une réponse temporelle d'un système discret.

1.3. Réponses fréquentielles des systèmes LTI

bode	- trace le diagramme de Bode d'un système continu.
dbode	- trace le diagramme de Bode d'un système discret.
nichols	- trace le lieu de Nichols (Black) d'un système continu.
dnichols	- trace le lieu de Nichols (Black) d'un système discret.
nyquist	- trace le lieu de Nyquist d'un système continu.
dnyquist	- trace le lieu de Nyquist d'un système discret.
evalfr	- évalue la réponse à une fréquence donnée.
margin	- donne les marges de gain et de phase d'un système continu ou discret.
freqresp	- réponse à une gamme de fréquence.
ngrids	- grille pour le lieu de Nichols.

1.4. Caractérisations temporelles et fréquentielles des systèmes LTI

damp	- donne le coefficient d'amortissement, la pulsation propre non-amortie (naturelle) et les pôles d'un système continu.
ddamp	- donne le coefficient d'amortissement, la pulsation propre non-amortie, et les pôles d'un système discret.
dcgain	- donne le gain statique d'un système continu.
ddcgain	- donne le gain statique d'un système discret.
pole	- donne les pôles d'un système.
tzero	- donne les zéros d'un système.
pzmap	- trace le graphe de la configuration pôles-zéros d'une fonction de transfert.
rlocus	- trace le lieu des racines (le lieu d'Evans).
sgrid	- affiche les courbes équi-amortissement du plan complexe 'p' (cette commande est à exécuter immédiatement après la commande rlocus).
zgrid	- affiche les courbes équi-amortissement du plan complexe 'z' (cette commande est à exécuter immédiatement après la commande rlocus).
rlocfind	- donne les valeurs des pôles et du gain correspondant sur le lieu d'Evans. (cette commande est à exécuter directement après la commande rlocus et sgrid).
minreal	- supprime des pôles composés par des zéros (simplification algébrique).
printsys	- affiche les fonctions de transfert comme fractions rationnelles en s ou en z.

1.5. Passage d'un modèle à l'autre

tf2zp	- passage de la forme polynomiale à la forme zéro-pôle et gain.
tf2ss	- passage de la forme polynomiale à la forme modèle d'état.
zp2tf	- passage de la forme zéro-pôle et gain à la forme polynomiale.
zp2ss	- passage de la forme zéro-pôle et gain à la forme modèle d'état.
ss2tf	- passage de la forme modèle d'état à la forme polynomiale.
ss2zp	- passage de la forme modèle d'état à la forme zéro-pôle et gain.
ss2ss	- passage de la forme modèle d'état à un autre modèle d'état avec matrice de transformation.

1.6. Discrétisation et conversion des systèmes LTI

c2d	- discrétise les modèles 'tf', 'ss' ou 'zpk' à temps continu.
d2c	- convertit les modèles 'tf', 'ss' ou 'zpk' à temps discret en temps continu.
d2d	- rééchantillonne un modèle à temps discret.

2. Principaux blocs de la boîte à outils "Simulink" utiles dans la régulation numérique

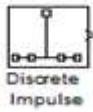
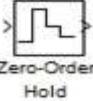
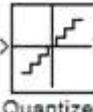
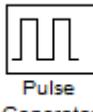
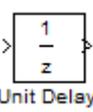
Nom du bloc	Emplacement	Description	Icône du bloc
Discrete Impulse	Signal Processing Blockset/Signal Processing Sources	Impulsion de Dirac discrète	
Zero-Order Hold	Discrete	Bloqueur d'ordre zéro	
First-Order Hold	Discrete	Bloqueur d'ordre un	
Quantizer	Discontinuities	Quantificateur	
Pulse Generator	Sources	Générateur d'impulsion	
Unit Delay	Discrete	Unité du retard	
Constant	Sources	Constante	
Step	Sources	Échelon	
Ramp	Sources	Rampe	
Sine Wave	Sources	Onde sinusoïdale	

Tableau (1) : Blocs utiles dans la régulation numérique

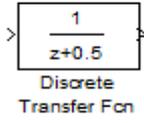
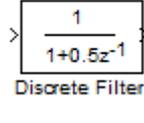
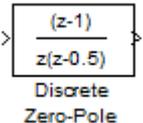
Discrete Transfert Fcn : Fonction de transfert discrète sous forme polynomiale en 'z'	Discrete Filter: Fonction de transfert discrète sous forme polynomiale en 'z ⁻¹ '	Discrete Zero-Pole: Fonction de transfert discrète sous forme Zéro-Pôle-Gain
		

Tableau (2) : Blocs utiles pour représenter les fonctions de transfert discrètes.
(Bibliothèque Discrete).

3. Principales bibliothèques de la boîte à outils "Simulink"

Nom de la bibliothèque	Descriptif
Sources	Générateurs de signaux, lecture de fichiers, etc.
Sinks	Blocs d'affichage ou de sortie.
Signal Routing	Routage du signal.
Math Operations	Opérations mathématiques.
Continuous	Blocs de modèles continus.
Discret	Blocs de modèles discrets.
Ports & Subsystems	Blocs permettant de réaliser des sous-systèmes.
Logic and Bit Operations	Blocs d'opérations logiques et binaires.
Discontinuities	Blocs modélisant les discontinuités.
Additional Math & Discrete	Blocs additionnels d'opérations mathématiques et blocs discrets.
User-Defined Functions	Fonctions définies par l'utilisateur.
Commonly Used Blocs	Blocs couramment utilisés.

Tableau (3) : Principales bibliothèques de la boîte à outils "Simulink"

4. Quelques commandes utiles en régulation numérique

- stem - commande permet de représenter les signaux échantillonnés.
- stairs - commande permet de représenter les signaux échantillonnés-bloqués par un *BOZ*.
- plot - trace une courbe en coordonnées linéaires.
- grid, grid on - met un grillage sur la courbe.
- grid off - masque le grillage de la courbe.
- hold on - permet de tracer plusieurs courbes dans la même fenêtre.
- hold off - annule la commande hold on.
- xlabel - place une étiquette sur l'axe x.
- ylabel - place une étiquette sur l'axe y.
- zlabel - place une étiquette sur l'axe z.

title	- met un titre sur la figure.
text	- place un texte sur la figure.
gtext	- place un texte sur une figure à l'aide de la souris.
legend	- ajoute une légende à la figure.
linspace	- Exemple : <code>linspace(a,b,n)</code> : crée un vecteur de n composantes régulièrement espacées entre les bornes a et b.
line	- permet de tracer une ligne.
axis	- définit manuellement les échelles des axes.
zoom	- permet d'agrandir une partie d'une figure.
figure	- crée une fenêtre graphique.

Notations

t : variable du temps continu
 k : variable du temps discret
 p : variable complexe de la transformée de Laplace
 z : variable complexe de la transformée en z
 z^{-1} : inverse de la variable z (retard d'un échantillon)
 Z : opérateur de la transformée en z
 \mathcal{L} : opérateur de la transformée de Laplace
 T, T_1, T_2 et T_3 : périodes d'échantillonnage
 $x(t)$ et $y(t)$: signaux en temps continu représentant généralement le signal d'entrée et le signal de sortie, respectivement
 $x^*(t)$ et $y^*(t)$: signaux échantillonnés issus respectivement de $x(t)$ et $y(t)$
 $x(k)$ et $y(k)$: signaux en temps discret représentant respectivement le signal d'entrée et le signal de sortie à l'instant d'échantillonnage kT
 $X(z)$ et $Y(z)$: représentent généralement la transformée en z des signaux d'entrée $x(k)$ et de sortie $y(k)$, respectivement
 $X(p)$ et $Y(p)$: représentent généralement la transformée de Laplace des signaux d'entrée $x(t)$ et de sortie $y(t)$, respectivement
 $X^*(p)$: transformée de Laplace du signal échantillonné $x^*(t)$
 $G(p)$: fonction de transfert continue d'un système linéaire continu G
 $G(z)$: fonction de transfert linéaire discrète (ou échantillonnée) du système G
 $B_0(p)$: fonction de transfert continue d'un bloqueur d'ordre zéro
 $B_1(p)$: fonction de transfert continue d'un bloqueur d'ordre un
 f : fréquence d'échantillonnage
 f_c : fréquence de coupure
 ω_c : pulsation de coupure
 ω_0 : pulsation propre
 τ : constante de temps (ou variable d'intégration en temps continu, ou temps de la tangente, ou retard, selon le contexte)
 ξ : coefficient d'amortissement
 $G_1(z), G_2(z), F_{d1}(z), F_{d2}(z), F_{d3}(z), F_{di}(z)$ et $F(z)$: fonctions de transfert discrètes
 $G_c(p)$ et $F_c(p)$: fonctions de transfert continues
 $G_s(z)$: fonction de transfert discrète résultante d'un montage en série
 $G_p(z)$: fonction de transfert discrète résultante d'un montage en parallèle
 $G_f(z)$: fonction de transfert discrète résultante d'un montage en rétroaction (feedback)
 T_{Min} : période d'échantillonnage minimale admissible
 T_{Max} : période d'échantillonnage maximale admissible
 R_0 : rayon du cercle unité dans le plan complexe z
 $Re(z)$ et $Re(p)$: parties réelles des variables complexes z et p , respectivement
 $Im(z)$ et $Im(p)$: parties imaginaires des variables complexes z et p , respectivement
 $p_i, p_{d1,2}, z_1$: pôles discrets
 p_1 et p_2 : pôles continus

z_1^* et p_1^* : sont les conjugués de z_1 et p_1 , respectivement
 r, c, β , et λ_0 : variables réelles
 $a_0, a_1, a_2, \dots, a_n, b_0, b_1, b_2, \dots, b_n, A, B, A_2$, et B_2 : coefficients réels constants
 i, n, m : variables entières
 ρ_1 et ρ_2 : variables complexes
 $N(z)$: polynôme numérateur
 $D(z)$: polynôme dénominateur (ou caractéristique)
 $\delta(k)$: impulsion de Dirac discrète
 $\delta(i)$: impulsion de Dirac à l'instant i
 $g(k)$: réponse impulsionnelle d'un système linéaire discret G
 $u(t)$: échelon unité en temps continu (ou loi de commande, selon le contexte)
 $u(k)$: échelon unité en temps discret (ou loi de commande, selon le contexte)
 K : gain ajustable (ou gain proportionnel, ou gain statique, selon le contexte)
 D : premier dépassement (ou premier pic)
 t_d : temps du premier pic
 t_m : temps de montée
 t_r : temps de réponse
 e_∞ : erreur statique (ou erreur en régime permanent)
 y_∞ : valeur finale du signal de sortie
 G_{dB} : gain exprimé en décibels (dB)
 φ : phase d'une fonction de transfert
 ω : pulsation variable (en rad/sec)
 j : unité imaginaire (telle que $j^2 = -1$)
 G_M (dB) : marge de gain en décibels
 φ_M (degré) : marge de phase en degré
 T_i : constante de temps de l'intégrale
 T_d , constante de temps de la dérivée
 $C_P(p)$: fonction de transfert d'un contrôleur proportionnel analogique
 $C_{PI}(p)$: fonction de transfert d'un contrôleur proportionnel-intégral analogique
 $C_{PID}(p)$: fonction de transfert d'un contrôleur proportionnel-intégral-dérivé analogique
 $e(t)$: erreur en temps continu
 $\frac{d}{dt}$: opérateur de dérivation continue
 $\frac{de}{dt}$: dérivée première de l'erreur en temps continu
 $E(z)$: erreur en temps discret (transformée en z de l'erreur)
 $U(z)$: commande en temps discret (transformée en z de la commande)
 L : retard d'un système en boucle ouverte
 K_u : gain ultime
 T_u : période ultime
 T_c : temps de la tangente
 a : pente de la tangente au point d'inflexion
 K_{osc} : gain critique mettant le système bouclé en oscillation entretenue

T_{osc} : période d'oscillation entretenue
 K_i : gain intégral
 K_d : gain dérivé
 s_1 et α : paramètre de filtrage de l'action dérivée dans un contrôleur PID numérique filtré
 R : résistance de l'induit
 L : inductance de l'induit
 K_c : constante de couple
 K_e : constante de force contre-électromotrice
 J : moment d'inertie du rotor
 $v(t)$: tension appliquée à l'induit
 $V(p)$: transformée de Laplace de la tension $v(t)$
 $i(t)$: courant circulant dans l'induit
 $I(p)$: transformée de Laplace du courant $i(t)$
 $\Omega(t)$: vitesse angulaire du moteur
 $\Omega(p)$: transformée de Laplace de la vitesse angulaire $\Omega(t)$
 f_{cem} : force contre-électromotrice
 $C_m(t)$: couple électromagnétique généré par le moteur
 $C_m(p)$: transformée de Laplace du couple électromagnétique $C_m(t)$
 $C_r(t)$: couple de charge
 $C_r(p)$: transformée de Laplace du couple de charge $C_r(t)$
 R_f : coefficient de frottement visqueux
 $r_0, r_1,$ et r_2 : paramètres réglables des contrôleurs PI/PID numériques
 $C(z)$: fonction de transfert discrète des contrôleurs PI/PID numériques exprimée en z
 $C(z^{-1})$: fonction de transfert discrète d'un contrôleur PID numérique exprimée en z^{-1}
 $G(z^{-1})$: fonction de transfert discrète du système exprimée en z^{-1}
 $R(z^{-1})$: polynôme R du contrôleur polynomiale RST numérique, associé à l'action de régulation
 $S(z^{-1})$: polynôme S du contrôleur polynomiale RST numérique, associé à l'action de correction du signal d'erreur
 $T_0(z^{-1})$: polynôme T du contrôleur polynomiale RST numérique, utilisé pour le suivi de la consigne
 $P(z^{-1})$: polynôme caractéristique imposé en fonction des spécifications désirées
 $B(z^{-1})$: polynôme numérateur en z^{-1}
 $A(z^{-1})$: polynôme dénominateur en z^{-1}
 $w(k)$: consigne discrète à l'instant d'échantillonnage kT
 $W(z)$: transformée en z de la consigne $w(k)$
 M : matrice de Sylvester
 ϕ : vecteur colonne contenant les paramètres de réglage du régulateur RST
 P : vecteur des coefficients du polynôme caractéristique imposé

Abréviations

LTI : Linear Time Invariant (système linéaire invariant dans le temps)
BOZ : Bloqueur d'ordre zéro
BOU : Bloqueur d'ordre un
TP : Travaux Pratiques
TPN01, ..., TPN8 : Travaux Pratiques Numéro 01 à 08
BO : Boucle ouverte
BF : Boucle fermée
P : contrôleur Proportionnel analogique (ou numérique)
PI : contrôleur Proportionnel-Intégral analogique (ou numérique)
PID : contrôleur Proportionnel-Intégral-Dérivé analogique (ou numérique)
MCC : moteur à courant continu
PIN1 : contrôleur PI numérique 1
PIN2 : contrôleur PI numérique 2
PIN3 : contrôleur PI numérique 3
RST : contrôleur Rétroaction-Suivi-Traitement numérique