TP automatique M1 SET
Asservissement du déplacement angulaire d'un moteur équipé d'un codeur incrémental

Objectif du TP

Asservir numériquement le déplacement angulaire du moteur en utilisant un codeur incrémental

Les différentes étapes:

TP1) Pilotage du hacheur par les sorties PWM1L et PWM1H du DSPIC30F4011

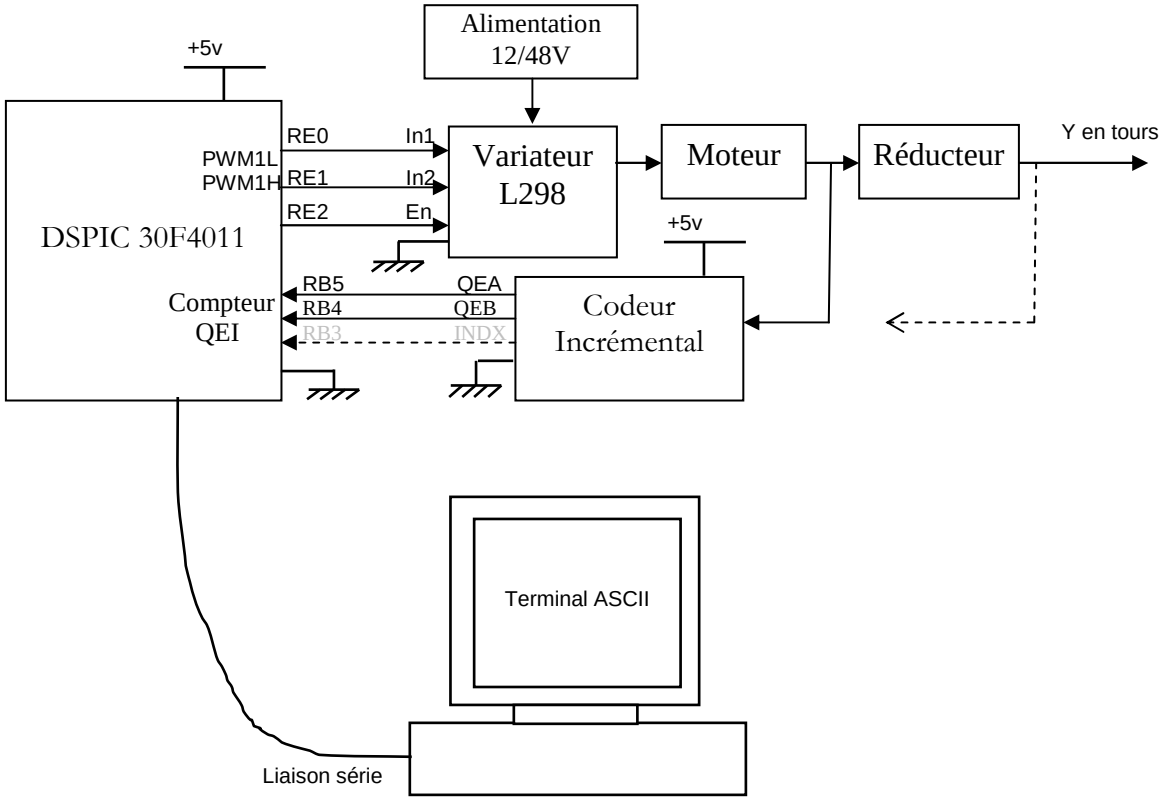
TP2) Utilisation du codeur incrémental et mise en œuvre du module QEI pour obtenir le déplacement angulaire et la vitesse du moteur

TP3)
- Identification (modélisation du dispositif à asservir)

TP4)
- Mise au point de la boucle d'asservissement de vitesse (consigne en tr/s)
méthode du modèle, calculs, simulation et tests sur le dispositif

TP5)
- Mise au point de la boucle d'asservissement du déplacement (consigne en tr)
Simulation et tests sur le dispositif.

Dispositif expérimental



TP1: Pilotage du hacheur (L298) par le DSPIC 30F4011

- Câbler le dispositif expérimental.

- Créer un nouveau projet avec les fichiers sources du TP1 pour le DSPIC 30F4011 quartz 8MHz
Options: "primary oscillator" "XT pll x 16" langage: "C30 Toolsuite" , le compiler et le charger dans la cible.

- Télécharger le document technique du circuit intégré L298. Le montage utilisé correspond au schéma d'application page 6. (http://www.datasheetcatalog.org/datasheet/SGSThomsonMicroelectronics/mXrqqxz.pdf)

Le circuit est prévu pour commander deux moteurs, on utilise ici un seul moteur:

- L'entrée "*EnA*" à *0* met hors tension le moteur, les 4 transistors sont alors bloqués

- Lorque "*EnA*" vaut 1 le moteur est alimenté:
  - en sens *1* si *In1 =0* et *In2=1*
  - en sens *2* si *In1=1* et *In2=0*

Pour obtenir une commande à vitesse réglable dans les deux sens de rotation:

- Le logiciel met à "1" l'entrée "*EnA*" et envoie deux signaux PWM complémentaires sur les entrées "*In1*" et "*In2*"

- le moteur réagit par rapport à la tension moyenne à ses bornes car la fréquence "PWM" utilisée est très grande (*25 kHz)*.

- Lorsque le rapport cyclique vaut *50%* ($r = \frac{Ton}{Tpwm} = 0.5$) la tension moyenne est nulle et le moteur est à l'arrêt.

- de 50% à 0 la tension moyenne est négative et de *50%* à *100%* elle est positive (ou l'inverse selon la convention choisie) c'est ce qui permet ici de changer la vitesse et le sens de rotation du moteur.

- plus le rapport cyclique s'éloigne de *50%* et plus la valeur absolue de la tension est élevée donc plus le moteur tourne vite.

- On démontre que $Umoy = Ualim.(2*r -1)$ avec le rapport cyclique $0 \leqslant r \leqslant 1$

1) Relever simultanément en utilisant les deux voies de l'oscilloscope les signaux *PWM1L* et *PWM1H* (broches *RE0* et *RE1*) pour:
a) *PDC1 = PTPER*
b) PDC1 = PTPER - 600
c) PDC1 = PTPER + 800

2) Relever avec un voltmètre en position "tension continue" (DC) la tension moyenne aux bornes du moteur pour huit valeurs de *PDC1* dans l'intervalle $0 \leqslant PDC1 \leqslant 2*PTPER$.

3) Tracer *Umoy* en fonction de *PDC1* et comparer à la formule théorique ci-dessous.

$Umoy = Ualim.(\frac{PDC1}{PTPER} -1)$ avec $0 \leqslant PDC1 \leqslant 2*PTPER$

4) Utiliser les informations du document sur les registres de la famille "30F" pour expliquer pourquoi *PDC1* doit varier de *0* à *2*PTPER* pour faire varier le rapport cyclique des sorties PWM1L et PWM1H respectivement de *0* à *1* et de *1* à *0*

5) Relever les valeurs de *PDC1* à la limite de la mise en rotation du moteur. Ces valeurs délimitent une "zone morte" elles seront utilisées pour la mise au point de l'asservissement.

TP2: Utilisation du codeur incrémental et mise en œuvre du module QEI

A) Etude du codeur

Télécharger la documentation technique du codeur équipant votre maquette.

A-1) Relever ses caractéristiques importantes: résolution (en degrés/top ou en tops/tours) , tension d'alimentation, type de sortie (PUSH-PULL ou collecteur ouvert), consommation.

ATTENTION : si les sorties sont à collecteur ouvert, il faut ajouter des résistances de rappel au +5V

A-2) Faire tourner lentement le moteur avec la fonction *test_pwm()* et relever à l'oscilloscope les signaux QEA et QEB du codeur (utiliser les deux voies de l'oscilloscope), que constatez-vous lorsqu'on inverse le sens de rotation?

A-3) Si le codeur possède une sortie "INDEX" , utiliser le mode 'single' de l'oscilloscope pour capturer et étudier ce signal.

Le signal "INDEX" est utilisé en général pour compter les tours entiers ou pour avoir une position angulaire de référence.

Il ne sera pas utilisé par la suite.

B) Etude du module QEI du DSPIC

Le module QEI (Quadrature Encodeur Interface) du DSPIC compte les fronts des signaux A et B provenant du codeur.
Le signe du décalage entre ces signaux indique au module s'il doit compter en montant ou en descendant, ceci permet de tenir compte automatiquement du sens de rotation.

Selon sa configuration ce module compte le fronts montants d'une ou deux entrées, cela permet d'améliorer la résolution du codeur d'un facteur 2 ou 4.

B-1) Etudier le programme et utiliser le document détaillé des registres des périphériques de la famille "30F" pour déterminer le mode de fonctionnement du module QEI.

B-2) Déterminer avec la documentation ou par un essai le nombre de tops "QEI" correspondant à un tour de l'arbre de sortie du dispositif.

B-3) Modifier la valeur TOPS_PAR_TOURS qui est affectée au registre MAXCNT et contrôler le fonctionnement du compteur lors du passage par cette valeur dans un sens puis dans l'autre.

B-4) Autoriser dans le programme d'initialisation l'interruption "override/underflow" qui se produit lorsque le compteur passe par sa valeur maximale ou par zéro selon le sens de rotation.

- Ecrire une routine pour compter les tours de codeur.
CONSEIL:
Tester le sens de rotation pour incrémenter ou décrémenter une variable globale qui contiendra ainsi le nombre entier de tours effectués.

- Modifier la routine *test_QEI()* pour afficher cette variable à côté de POSCNT

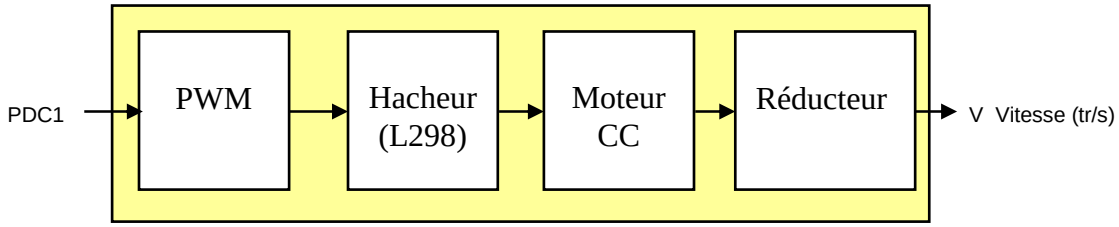
- Vérifier le bon fonctionnement pour les deux sens de rotation.

B-5) Ecrire et tester une fonction qui retourne un nombre en virgule flottante (float) donnant la position de l'arbre de sortie du réducteur en nombre de tours (par exemple : 1,25 tr représente un tour et un quart)

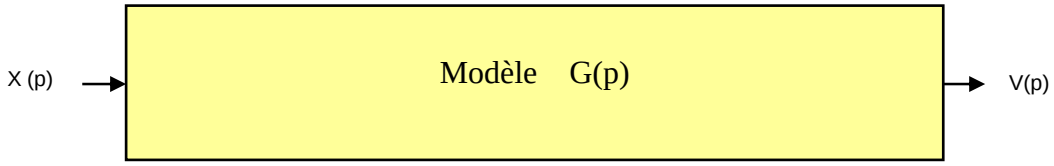
TP3 : Modélisation du dispositif (Identification)

A) Introduction et présentation de la méthode d'identification

Pour optimiser l'asservissement de vitesse du dispositif on a besoin du modèle mathématique du dispositif à contrôler, ici l'ensemble formé par le générateur PWM, le hacheur, le moteur et le réducteur



Le dispositif à contrôler est supposé linéaire, continu et invariant dans le temps, on peut alors utiliser la transformée de Laplace G(p) pour le modéliser (On remplace PDC1 par x = PDC1-1279 pour la linéarité)



A-1 Deux méthodes peuvent être utilisées :

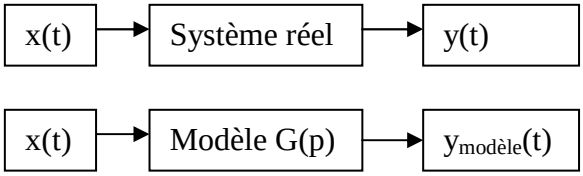
1-Mise en équation du fonctionnement physique.
La mise en équation directe, nécessite une connaissance approfondie du dispositif. Elle est difficile à mettre en œuvre pour les systèmes complexes (frottement, inertie, documentation incomplète).

2-Etude de la réponse du procédé à un signal d'entrée simple.
Cette méthode expérimentale consiste à faire correspondre la réponse d'un modèle mathématique à celle du procédé.
Les signaux d'entrée les plus utilisés sont les impulsions, les échelons, les signaux aléatoires et les signaux sinusoïdaux.

a) La réponse impulsionnelle fournit directement la fonction de transfert, mais on ne peut pas réaliser physiquement une impulsion de Dirac, aussi utilise-t-on une impulsion de durée très courte et d'amplitude finie. Les résultats sont en général peu précis car la réponse est de faible amplitude.
La réponse impulsionnelle peut cependant être obtenue de façon plus précise par intercorrélation entrée-sortie pour un signal pseudo-aléatoire d'entrée.

b) La réponse indicielle (échelon) est très souvent utilisée car elle est simple à obtenir et à interpréter.
De nombreuses méthodes d'identification se basent sur cette réponse ; notamment les méthodes de Strejc, Broïda,... qui permettent de choisir rapidement les coefficients d'un correcteur PID.

A-2) Dans ce TP on utilise une modélisation basée sur la réponse indicielle

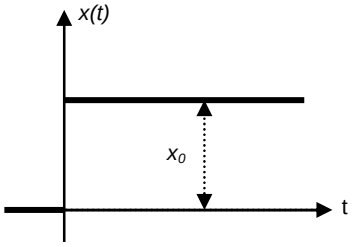


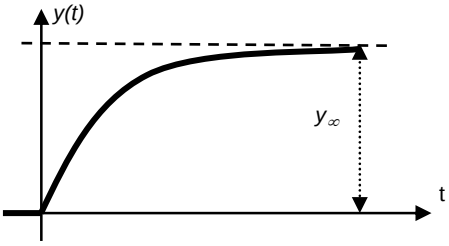
Le procédé à modéliser ici est par nature stable (à une entrée PDC1 finie correspond une vitesse de sortie finie)
La fonction de transfert ne comporte donc pas de terme intégrateur, elle est de la forme:

$$G(p) = \frac{A}{a_n p^n + \ldots a_2 p^2 + a_1 p + 1}$$

L'identification consiste à calculer les paramètres A, a_1, a_2…a_n du modèle pour faire correspondre les réponses y(t) et $y_{modèle}(t)$

On applique en entrée un saut d'amplitude x_0 , et on enregistre la réponse $y(t)$.





$$X(p) = \frac{x_0}{p} \qquad Y(p) = G(p).X(p) = \frac{A}{a_n p^n + \ldots a_2 p^2 + a_1 p + 1} \cdot \frac{x_0}{p}$$

A-2-1) Calcul de A:

On utilise le théorème de la valeur finale :

$$\lim_{t \to +\infty} y(t) = \lim_{p \to 0} p.Y(p) = A.x_0 \quad \Rightarrow \quad \boxed{A = \frac{y_\infty}{x_0}}$$

En pratique, on relève la valeur de y lorsque la sortie s'est stabilisée (ou la moyenne des dernières valeurs)

A-2-2) Calcul de a_1 :

Soit $d_1(t) = x_0 - \frac{y(t)}{A}$

$$D_1(p) = \frac{x_0}{p}\left(1 - \frac{1}{a_n p^n + \ldots a_2 p^2 + a_1 p + 1}\right) = \frac{x_0}{p} \cdot \frac{a_n p^n + \ldots a_2 p^2 + a_1 p}{a_n p^n + \ldots a_2 p^2 + a_1 p + 1}$$

$$\lim_{p \to 0} D1(p) = a.x_0$$

En remarquant que $\lim_{p \to 0} F(p) = \lim_{p \to 0} p\left(\frac{F(p)}{p}\right) = \lim_{t \to +\infty} \int_0^t f(\tau).d\tau$

On en déduit que $\lim_{p \to 0} D1(p) = \lim_{t \to +\infty} \int_0^t d1(\tau)\, d\tau$

D'où : $$\boxed{a_1 = \lim_{t \to +\infty} \frac{\int_0^t d1(\tau)\, d\tau}{x_0}}$$

A-2-3) Calcul de a_2 :

soit $d_2(t) = \int_0^t d1(\tau)\, d\tau - a_1 . \frac{y(t)}{A}$

On montre comme précédemment que $$\boxed{a_2 = \lim_{t \to +\infty} \frac{\int_0^t d2(\tau)\, d\tau}{x_0}}$$

A-2-4) On généralise le résultat précédent pour calculer les a_k

REMARQUES:

- En pratique on se limitera à l'ordre 1 dans ce TP , a_1 est la constante de temps du dispositif, on peut l'évaluer graphiquement.

B) Mise en œuvre de la méthode d'identification

- Créer un nouveau projet avec les fichiers sources du TP3 pour le DSPIC 30F4011 quartz 8MHz
Options: "primary oscillator" "XT pll x 16" langage: "C30 Toolsuite" , le compiler et le charger dans la cible.

- La routine *reponse()* demande le nombre d'échantillons et la valeur de *PDC1*, autorise l'interruption du Timer2, et attend.

- La routine d'interruption Timer2 lit le compteur QEI (POSCNT) et envoie automatiquement le résultat au port série.

- Lorsque le nombre d'échantillons demandé est atteint, la routine d'interruption se désactive et arrête le moteur.

B-1) Utiliser hyperterminal pour capturer la réponse à un échelon de commande PDC1

Faire plusieurs essais, et indiquer dans le nom de fichier les valeurs:

- de l'échelon de commande x_0 = ***PDC1-1279*** =
- de la fréquence d'échantillonnage F***e*** =
- et du nombre de points enregistrés ***N*** =

Vous choisirez par la suite un fichier qui met bien en évidence le régime transitoire et le début du régime permanent où la vitesse se stabilise.

B-2) Identification du procédé à l'aide d'un logiciel de calcul numérique (matlab)

Le fichier obtenu précédemment contient *N* échantillons du compteur *POSCNT* , ils représentent l'image numérique du déplacement angulaire en impulsions codeur soit Yd.

B-2-1 Afficher le graphique *Yd(t)* des valeurs brutes obtenues en fonction du temps en utilisant l'application matlab "affiche_capture.m" fournie.

Selon la durée de l'enregistrement, il se peut que *Yd* atteigne *MAXCNT*, dans ce cas le compteur repasse brutalement à *0* au $n^{ième}$ échantillon il faut alors ajouter *MAXCNT* aux valeurs suivantes de *Yd*:

*Yd (n : N) = Yd (n : N) + MAXCNT * ones(N – n +1 , 1);*

B-2-2 Modifier l'application matlab pour obtenir le déplacement angulaire de l'arbre de sortie *Yd en tours*.

Afficher le graphique *Yd(t).*

B-2-3 Calculer la vitesse *V* en *tr/s* avec la formule approchée:

$$V(i) = \frac{Yd(i+1) - Yd(i-1)}{2*Te} \quad \text{pour} \quad i \text{ allant de } 2 \text{ à } N-1$$

$V(1) = 0$ la première valeur vaut 0 car le moteur est à l'arrêt.

$$V(N) = \frac{Yd(N) - Yd(N-1)}{Te}$$

Afficher le graphique *V(t)* :
Utiliser ***plot…*** puis ***hold on*** pour conserver le graphique pour la suite (utile pour superposer plusieurs courbes)

B-2-4 Calcul des paramètres *A et* a_1 du modèle *G(p) au premier ordre*

ATTENTION:
La sortie y du modèle est la vitesse V en tr/s et l'entrée x_0 *la variation de commande "PDC1- 1279"*

Calcul de *A (voir le paragraphe A-2-1):*

On assimilera $y(\infty)$ à la moyenne de *y* sur les derniers échantillons (*10* ou plus, suivant l'allure de la courbe).
mean(y(n_1:n_2)) donne la moyenne des valeurs de $y(n_1)….y(n_2)$

Calcul de *(voir les paragraphes A-2-2):*

Les formules établies doivent être discrétisées :
- les intégrales sont remplacées par des sommes
- l'intervalle de calcul est limité aux *N* échantillons
- *dt* et $d\tau$ sont remplacés par *Te* .

B-2-4-1 Donner la formule discrétisées de a_1

B-2-4-2 Ecrire, dans un fichier matlab, les instructions pour calculer A et *a1*

B-2-4-3 Donner les valeurs numériques obtenues pour *A et* a_1

C) Simulation de la réponse indicielle du modèle et comparaison avec *V(t)*

C-1) Utilisation de matlab

La commande ***step****(numérateur,dénominateur, t)* affiche la réponse à l'échelon **unité** d'un système dont la fonction de transfert est un rapport de deux polynômes.

Utiliser la commande *step* pour afficher la réponse à un échelon d'amplitude x_0

a) Pour le modèle du premier ordre $G(p) = \frac{A}{a_1 p+1}$

c) Comparer la courbe obtenue à la réponse *y(t) (hold on pour conserver et superposer les courbes)*

D) Utilisation de simulink

- Entrer la commande ***simulink*** (ou cliquer sur l'icône)
- Copier dans la feuille de travail le processus $\frac{1}{1+p}$ de la bibliothèque 'linear' . Cliquer sur le bouton droit et remplacer les paramètres par ceux du modèle du second ordre précédent.

- Copier l'échelon de la bibliothèque 'sources' vers la feuille de travail. Modifier les paramètres de l'échelon. Connecter l'échelon à l'entrée du modèle.

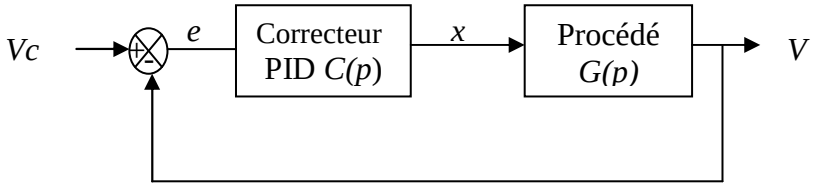
- Placer l'oscilloscope, qui se trouve dans 'sinks', en sortie du modèle.

- Paramétrer la durée de la simulation, le pas des calculs et cliquer sur **start** pour lancer la simulation. Cliquer 2 fois sur l'oscilloscope pour voir le résultat.

Comparer aux résultats précédents

TP4- Mise au point de la boucle d'asservissement de vitesse
Détermination des paramètres du correcteur et simulation en boucle fermée

INTRODUCTION



Vc est la vitesse souhaitée, V la vitesse réelle, e=Vc-V l'erreur et x la commande (PDC1-1279)
G(p) le modèle du dispositif à contrôler obtenu dans le TP3
C(p) la fonction de transfert du correcteur à mettre au point dans ce TP.
L'objectif est d'obtenir une vitesse de sortie du dispositif la plus proche possible de la consigne.

Cette étude comporte quatre parties:

A) Calcul des paramètres d'un correcteur destiné à contrôler la vitesse de rotation du moteur par l'une des trois méthodes présentées ci-dessous A-1, A-2 et A-3.

B) Simulation du fonctionnement analogique du système en boucle fermée en utilisant simulink .
Choix du correcteur qui donne les meilleurs résultats.

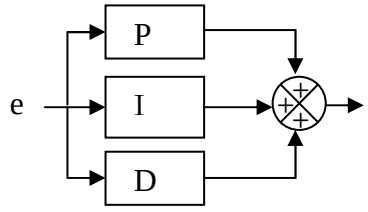
C) Numérisation du correcteur, simulation du dispositif avec le correcteur numérique pour vérifier son fonctionnement.

D) Implantation du correcteur numérique dans le DSPIC et vérification de ses performances.

RAPPELS SUR LES CORRECTEURS PID

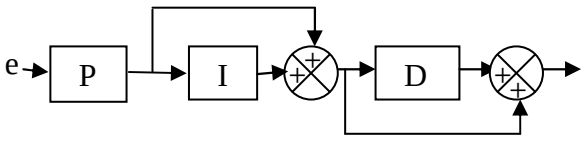
Il y a existe plusieurs type de correcteur P.I.D. selon la façon d'associer les paramètres P, I et D .

P.I.D. parallèle :



$$x = K.e + \frac{1}{T_i}.\int_0^t e.dt + T_d.\frac{de}{dt} \quad \Rightarrow \quad C(p) = K + \frac{1}{Ti\,p} + Td\,p$$

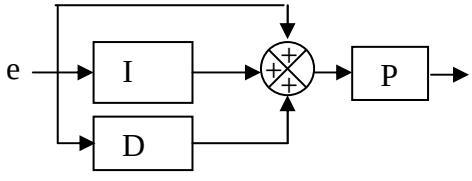
P.I.D. série



$$x = K\left((1+\frac{T_d}{T_i}).e + \frac{1}{T_i}.\int_0^t e.dt + T_d.\frac{de}{dt}\right)$$

$$C(p) = K\left(1 + \frac{Td}{Ti}\right) + \frac{1}{Ti\,p} + Td\,p$$

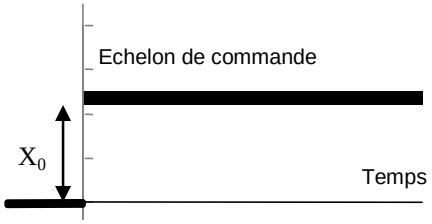
P.I.D. mixte



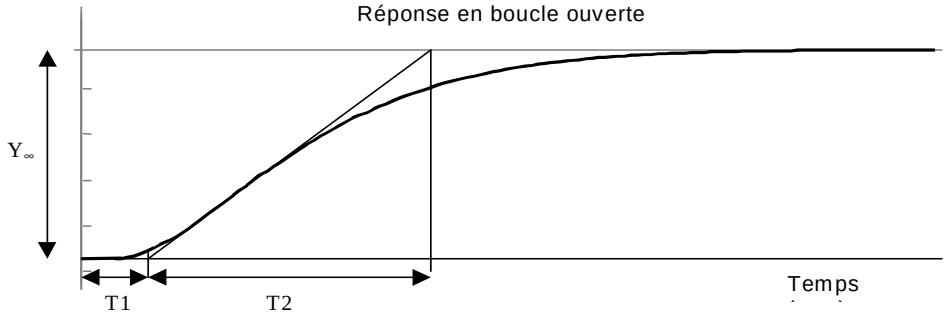
$$x = K\left(e + \frac{1}{T_i}.\int_0^t e.dt + T_d.\frac{de}{dt}\right) \Rightarrow C(p) = K\left(1 + \frac{1}{Ti\,p} + Td\,p\right)$$

A-1) Détermination des paramètres du correcteur P.I.D. par la méthode de Chien-Hrones-Reswick. (pas utilisée dans ce TP)

Cette méthode est applicable aux procédés stables et se base sur la réponse indicielle du procédé à contrôler. On trace la tangente au point d'inflexion de la réponse et on mesure *T1* et *T2*.
Les paramètres du correcteur se calculent à l'aide du tableau ci dessous.

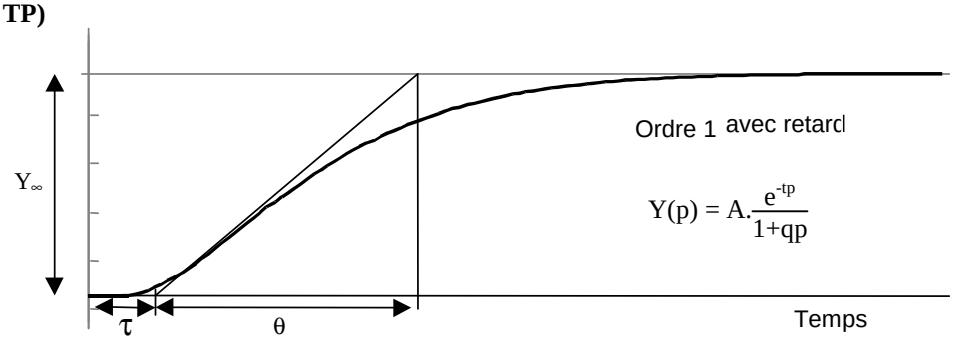


	Paramètres	Réponse apériodique	dépassement D < 20%
P	K	$0.3\frac{T_2}{T_1}$	$0.7\frac{T_2}{T_1}$
PI	K	$0.35\frac{T_2}{T_1}$	$0.6\frac{T_2}{T_1}$
	Ti	$1.2T_2$	T_2
PID série	K	$0.6\frac{T_2}{T_1}$	$0.95\frac{T_2}{T_1}$
	Ti	T_2	$1.35T_2$
	Td	$0.5T_1$	$0.47T_1$



A-1-1 Calculer K et Ti pour un correcteur PI par cette méthode

A-2) Détermination des paramètres du correcteur P.I.D. par la méthode de Broïda. (pas utilisée dans ce TP)



Dans la méthode de Broïda on approxime la réponse indicielle à une réponse du premier ordre avec un retard.

On relève les temps t_1 et t_2 correspondants à $y_1 = 0.28.y_\infty$ et $y_2 = 0.40.y_\infty$

On calcule ensuite $A = \frac{y_\infty}{x_0}$,

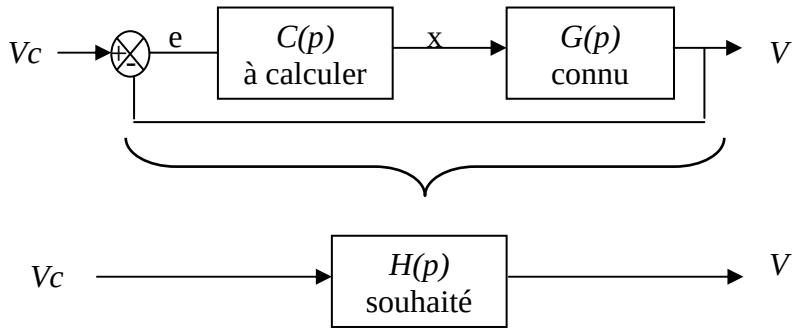
$\tau = 2.8t_1 - 1.8t_2$ et $\theta = 5.5(t2 - t1)$

et on en déduit les paramètres du correcteur avec le tableau ci-contre.

	Paramètres	Dépassement D < 20%
P	K	$\frac{0.8\theta}{A\tau}$
PI	K	$\frac{0.8\theta}{A\tau}$
	Ti	θ
PID série	K	$\frac{0.8\theta}{A\tau}$
	Ti	θ
	Td	0.4τ

A-2-1 Calculer K et Ti pour un correcteur PI par cette méthode

A-3) Calcul du correcteur par la méthode du modèle (à utiliser dans ce TP)



On calcule le correcteur pour obtenir une fonction de transfert globale *H(p)* donnée.

A-3-1) Montrer que $C = \frac{1}{G(\frac{1}{H} - 1)}$

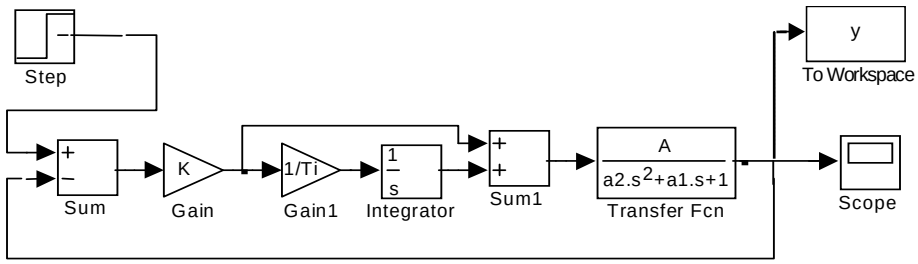
A-3-2) En utilisant $G = \frac{A}{1 + a\,p}$ du modèle à l'ordre 1 calculé dans le TP3 et en prenant pour modèle à atteindre $H = \frac{1}{1 + a\,p}$, montrer que le correcteur est de type proportionnel intégral série:

C(p) = K $(1+\frac{1}{Ti\,p})$, calculer les valeurs de K et Ti ?

B) Simulation, choix du meilleur correcteur.

B-1 Utiliser *simulink* pour simuler en analogique la réponse du système bouclé à un échelon de consigne V_c = 1tr/s ou 10 *tr/s* selon les dispositifs.

- Utiliser la méthode A-3) pour calculer les paramètre du correcteur "PI" (K et Ti)
- Les résultats peuvent être enregistrés dans un vecteur y pour une utilisation avec matlab. (ToWorkspace)



B-2 Simuler la réponse à une rampe de consigne *v(t) = 0.1.t* de durée *5s*

B-3 Donner la fonction de transfert symbolique *T(p)=C(p).G(p)* de l'ensemble correcteur-procédé en boucle ouverte sous forme d'un rapport de polynômes : $T(p) = \frac{N(p)}{D(p)}$

~~B-4 Utiliser les commandes ***Bode*** et ***Nyquist*** de matlab pour tracer les diagrammes du même nom. En déduire la marge de gain et la marge de phase.~~

~~B-5 Utiliser la commande ***cloop*** ou ***feedback*** pour obtenir la fonction de transfert en boucle fermée.~~
~~Calculer avec la commande ***roots*** ses zéros et ses pôles.(On peut également utiliser ***tf2zp***)~~
~~Commenter les résultats.~~

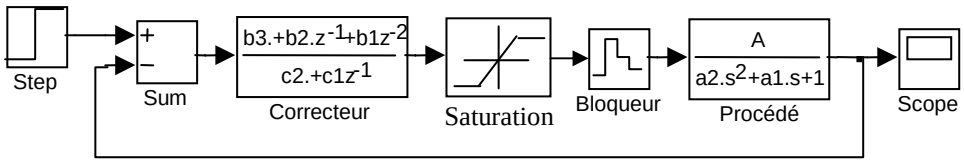
C) Numérisation du correcteur.

C-1 Ecrire sous forme discrète l'équation temporelle du correcteur PI série et en déduire une équation de récurrence donnant *x(n)* en fonction de *e(n)… x(n-1)….*
C'est cette équation qui sera utilisée dans le programme du DSPIC.

C-2 Donner la transformée en *z* du correcteur en fonction de *K*, T_i, z^{-1}….

C-3) Utiliser simulink pour tester le comportement du dispositif avec les paramètres calculés précédemment avec une période d'échantillonnage *Te = ... (utiliser la période d'échantillonnage du TP2)*

Le bloc de saturation doit tenir compte des valeurs extrêmes de x = (PDC1-1279).



D) Implantation du correcteur dans le DSPIC 30F4011 et validation des algorithmes sur le dispositif réel

D-1 Ecrire un programme en C pour le DSPIC qui:

- demande en boucle la consigne de vitesse en *tr/s* (programme principal)

- calcule la commande x(n) et envoie PDC1(n) dans le registre de génération PWM tous les *Te* (routine d'interruption Timer2)

- vérifier que la vitesse demandée est obtenue, sinon évaluer l'erreur et chercher une explication.

- Modifier le programme pour tenir compte de la zone morte (voir TP1 -5)

D-2) Etudier expérimentalement

a) l'évolution de la vitesse lors d'une augmentation ou diminution brutale de la charge ?
(On peut freiner l'arbre manuellement avec précaution)

b) la réponse à un échelon de consigne.

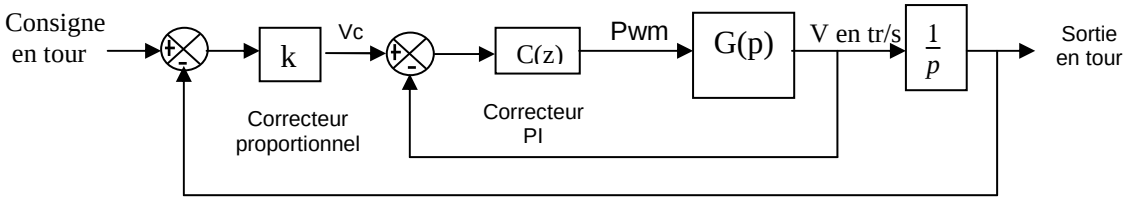
D-3) Pour obtenir un démarrage progressif, ajouter au programme une fonction pour fabriquer une rampe vitesse. La fonction doit demander la vitesse à atteindre et la durée de la rampe.
Vérifier.

Faire valider par l'enseignant votre travail lorsque la régulation fonctionne correctement.

.

TP5) Mise au point de la boucle d'asservissement du déplacement (consigne en tr)

Principe : L'écart sur la position, multiplié par la constante k, sert de consigne de vitesse.
Ceci permet d'adapter automatiquement la vitesse à la position, par exemple de ralentir dès que le système s'approche de la consigne.



A) Simuler la réponse à un échelon de déplacement pour différentes valeurs de k.
Déterminer k qui donne une réponse amortie avec un seul dépassement.

B) Ecrire et tester le programme avec le dispositif réel, faire valider le travail réalisé lorsque ça fonctionne.