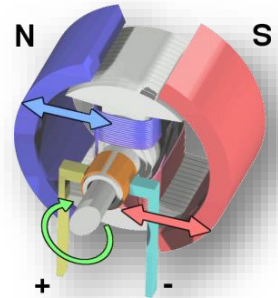


## Travaux Pratiques CORRECTEURS NUMERIQUES

### ACTIVITE 1 : Simulation d'un Moteur à Courant Continu (MCC)



On rappelle les équations du moteur à courant continu :

(1)	$u(t) = e(t) + Ri(t) + L \frac{di(t)}{dt}$	Equations électriques du moteur à courant continu
(2)	$e(t) = k_e \omega(t)$	
(3)	$c_m(t) = k_c i(t)$	
(4)	$c_f(t) = f \omega(t)$	Couple de frottement fluide
(5)	$c_m(t) - c_f(t) - c_r(t) = J \frac{d\omega(t)}{dt}$	Principe fondamental de la dynamique

- $u(t)$  : Tension d'entrée aux bornes du moteur (V)
- $e(t)$  : Force contre électromotrice (V)
- $i(t)$  : Intensité (A)
- $\omega(t)$  : Vitesse de rotation du moteur ( $rad.s^{-1}$ )
- $c_m(t)$  : Couple moteur ( $N.m$ )
- $c_f(t)$  : Couple de frottement ( $N.m$ )
- $c_r(t)$  : Couple résistant ( $N.m$ )
- $L$  : Inductance de la bobine (H)
- $f$  : Coefficient de frottement visqueux ( $N.m.s.rad^{-1}$ )
- $J$  : Inertie équivalente en rotation de l'arbre moteur ( $Kg.m^2$ )
- $R$  : Résistance électrique du moteur ( $\Omega$ )
- $k_e$  : Constante de force électromotrice ( $V.rad^{-1}.s$ )
- $k_c$  : Constante de couple ( $N.m.A^{-1}$ )
- 



$$R = 0,45 \, \Omega \quad ; \quad L = 0,0046 \, H \quad ; \quad k_e = 0,169 \, V.rad^{-1}.s \quad ; \quad k_c = 0,17 \, N.m.A^{-1}$$

$$f = 0,01 \, N.m.s.rad^{-1} \quad ; \quad J = 0,01 \, Kg.m^2 \quad ; \quad c_r = 0 \, N.m$$



Exprimer  $\frac{d(i(t))}{dt}$  et  $\frac{d(\omega(t))}{dt}$  en fonction des autres grandeurs et des différentes constantes. Mettre en forme ces équations pour une résolution d'Euler (pour l'intensité et la vitesse de rotation du moteur).



Compléter le code *EULER\_MCC\_ELEVES.py* afin de programmer la résolution Euler explicite des équations pour un échelon de tension de 10 V sur 1 secondes avec 1000 points.



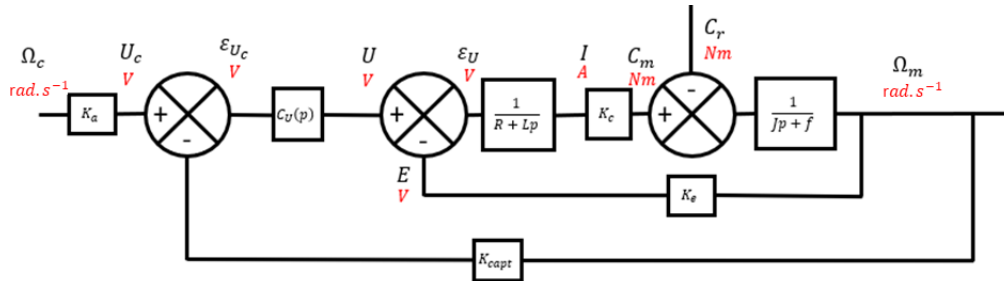
Commenter l'allure des courbes donnant la vitesse du moteur et de son intensité en fonction du temps.



Réaliser le modèle X-COS du MCC (compléter le fichier *Scilab MCC\_ELEVES.zcos*) et comparer les résultats obtenus.

## ACTIVITE 2 : Simulation numérique de l'asservissement du MCC

On souhaite maintenant mettre en place un asservissement en vitesse de rotation du moteur.  
On propose le schéma bloc suivant :



On donne :

$$\omega_c = 1 \text{ rad.s}^{-1} ; \quad K_{capt} = 10 \text{ V.s.rad}^{-1} ; \quad K_a = K_{capt} ; \quad C(p) = 1$$

A chaque pas de temps  $dt$ , l'asservissement doit :

- Mettre à jour la liste  $V=[i,w]$  de laquelle le moteur va partir
- Mettre à jour le temps
- Calculer le retour  $r$  et l'écart  $eps$
- Mettre à jour la tension  $u$  (variable globale pour MCC) à bien mettre AVANT MCC
- Stocker les données associées à la tension  $u$  imposée ( $t, u$ ) dans les listes  $Ltu$  et  $Lu$
- Simuler l'évolution du MCC avec cette consigne  $u$  depuis l'état  $V$  entre les temps  $t$  et  $t+dt$  avec un pas de temps  $dt_s = dt = 2.10^{-4} \text{ s}$
- Compléter les listes  $Lt, Li, Lw$  des données de simulations obtenues par paquets par la simulation du MCC  $lt, li$  et  $lw$



Ouvrir le code Python `ASSERVISSEMENT_MCC_ELEVES.py` et identifier les différentes étapes que doit réaliser l'asservissement.



Compléter le code permettant de réaliser la simulation de l'asservissement sur 0,2 secondes (compléter le code `ASSERVISSEMENT_MCC_ELEVES.py`).



Commenter l'allure des courbes donnant la vitesse du moteur et de son intensité en fonction du temps.



Comparer l'évolution de l'intensité et de la vitesse du moteur au cours du temps aux résultats obtenus par une simulation du même système réalisée sur X-COS (compléter le fichier `Scilab ASSERVISSEMENT_MCC.zcos`).

### ACTIVITE 3 : Correcteurs numériques

Pour simuler un asservissement sans implémenter de correcteur numérique dans un système réel, il est possible de rajouter une 3<sup>ème</sup> équation représentant le correcteur.

Cependant, l'objectif ici est d'apprendre à implémenter un correcteur numérique sur un système réel et de le vérifier avec une simulation.

Pour une résolution numérique optimale, on fixe à nouveau  $dt = dts = T_e$ .

A l'itération  $k$  de la boucle d'asservissement, on appelle :

- $e_k$  l'écart au comparateur
- $u_k$  la tension de commande du moteur élaborée par le correcteur

Sans correcteur,  $u_k = e_k$ .



**Il faudra dans votre code noter  $u$ , et non  $uk$ , car la fonction d'Euler utilise la tension  $u$  comme variable globale.**

On propose les correcteurs suivants :

P	PI	Avance de phase
$K_p$	$K_p + \frac{K_i}{p}$	$\frac{1 + aTp}{1 + Tp}$ avec $a > 1$



A l'aide du fichier *Scilab ASSERVISSEMENT\_NUMERIQUE\_BODE.zcos*, justifier le choix de la valeur de  $T_e$ .



En utilisant la méthode des équations aux différences  $\left( \frac{d(s(t))}{dt} \approx \frac{s(t) - s(t - T_e)}{T_e} \right)$ , déterminer pour ces trois correcteurs l'expression de  $u_k$  en fonction de  $u_{k-1}$ ,  $e_k$  et  $e_{k-1}$ .



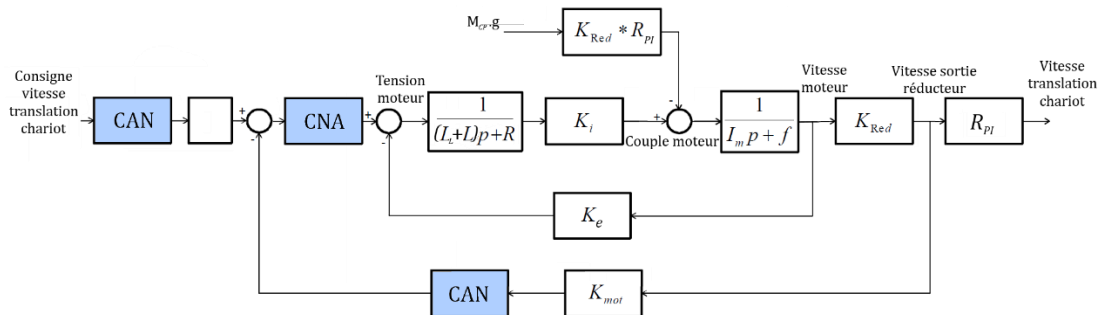
Proposer les trois fonctions python *CP*, *CPI* et *CAP* des correcteurs numériques étudiés en précisant les arguments nécessaires.



En utilisant le code de l'ACTIVITE 2, ajouter ces 3 correcteurs à l'asservissement numérique réalisé et vérifier leur fonctionnement avec *X-COS*. Modifier également les calculs de l'écart  $e_k$  et de la tension  $u$ . Les coefficients des correcteurs seront réglés afin d'obtenir une marge de phase de  $45^\circ$ .

## ACTIVITE 4 : Modéliser un Convertisseur Analogique – Numérique (CAN)

La partie commande d'un asservissement étant numérique, le schéma-bloc intègre en pratique des CAN et des CNA, comme représenté sur le schéma suivant :

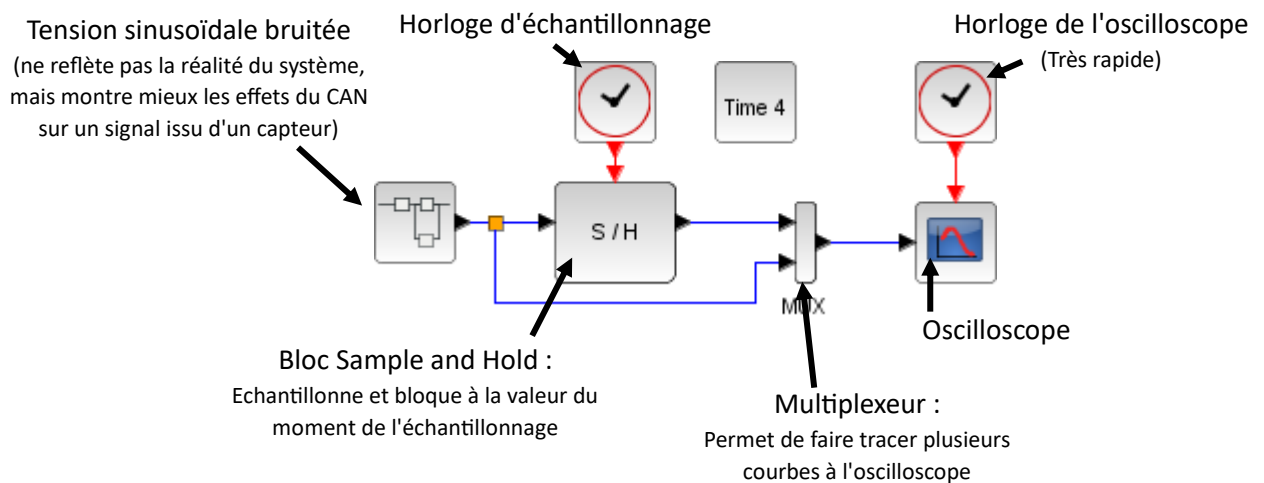


L'étude portera uniquement sur le fonctionnement d'un convertisseur analogique numérique (CAN), qui réalise l'échantillonnage et la quantification d'un signal analogique.

## Echantillonnage



Ouvrir le fichier **CAN\_ELEVES.zcos** et visualiser le schéma-bloc suivant.



La tension d'entrée sinusoïdale proposée a une pulsation de  $12 \text{ rad/s}$ . Le signal d'entrée est bruité afin de refléter le signal issu d'un capteur.



Calculer la période d'échantillonnage pour mesurer 10 points par période.



Modifier l'horloge d'échantillonnage avec la valeur obtenue, puis lancer une simulation.

Les courbes obtenues correspondent au signal sinusoïdal d'entrée, et au signal échantillonné.



Modifier la période d'échantillonnage vers des valeurs plus ou moins importantes.



Quelle est l'influence d'une période d'échantillonnage :

- Trop faible
- Trop importante
- Mal adaptée (0,3 s par exemple)



Expliquer pourquoi on parle "d'échantillonneur bloqueur" (bloc Sample and Hold) ?  
Expliquer pourquoi le bloqueur utilisé ici est d'ordre 0.

## Quantification

Nous allons maintenant ajouter la quantification (discrétisation en amplitude) au signal précédent.



Configurer la période d'échantillonnage à 0,05 s (valeur proche de 10 éch/s).



Ajouter un bloc de quantification au schéma-bloc précédent.

Remarque : Le bloc de quantification se trouve dans la rubrique *Palettes/Fonctions discontinues* du navigateur de palettes. Choisir une amplitude de quantification de 0,5 V, et laisser la variable *Quantization type* à 1 (arrondi à l'incrément le plus proche).



Modifier l'amplitude de quantification pour voir son influence.



Quelle est l'influence d'une amplitude de quantification :

- Trop faible
- Trop importante



Paramétrer à nouveau l'amplitude de quantification à 0,5 V.

Nous venons de réaliser un convertisseur analogique numérique : chaque valeur quantifiée est ensuite convertie en binaire et communiquée à la partie commande. La valeur binaire transmise change à chaque nouvel échantillonnage.

## Effet du CAN sur le signal

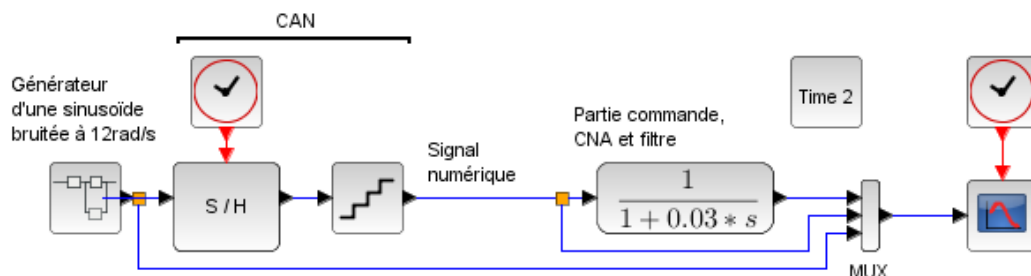
On suppose que :

- Le temps de traitement du signal par la partie commande est négligeable.
- Le délai de réponse du CNA en sortie de partie commande est négligeable.

En sortie du CNA, se trouve un filtre de fonction de transfert  $F(p) = \frac{1}{1+0.03.p}$  permettant un lissage du signal obtenu. Pour mieux visualiser les effets du CAN étudié précédemment, le gain statique de la partie commande a été choisi unitaire.



Mettre en place ce filtre en sortie du CAN, comme sur la figure ci-dessous. On note que l'on a ajouté une voie à l'oscilloscope (double cliquer dessus puis changer le nombre de voies).



Relancer la simulation avec ce schéma-bloc.



Quel est le principal effet qu'a eu la conversion analogique numérique ? Donner une valeur caractéristique de cet effet. A quoi correspond cette valeur ?



Conclure quant à l'effet d'une conversion analogique → numérique utilisant un bloqueur d'ordre 0.