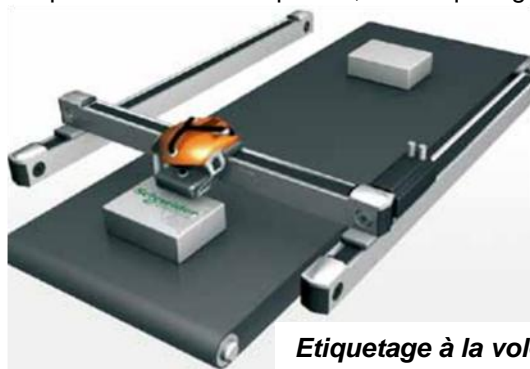
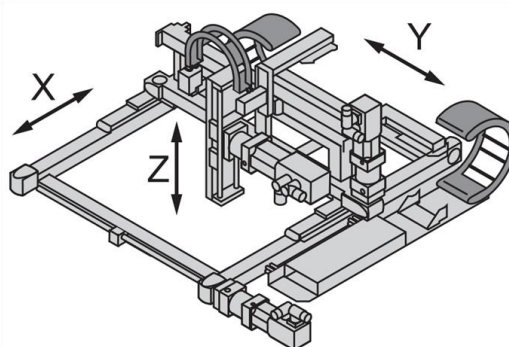


### Axe de robot Control'X

Le système industriel dont est extrait Control'X est un robot portique 3 axes Lexium Max R du constructeur Schneider Electric. Ce robot portique est constitué :

- D'un axe portique double Lexium MAX S assurant un déplacement selon la direction X
- D'un axe portique double Lexium MAX H assurant un déplacement selon la direction Y.
- D'un axe Cantilever Lexium CAS 4 ou Lexium CAS 3 assurant un déplacement selon la direction Z.

Les applications typiques de ces portiques relèvent du "pick and place", de la manutention, de la palettisation, du tri, de l'inspection/mesure de pièces, de l'étiquetage...



*Etiquetage à la volée*

**Problématique**

Suite à des problèmes de positionnement de l'axe, la société Schneider Electric vous demande de régler au mieux le correcteur PID de l'asservissement de position à partir d'essais , puis de concevoir l'algorithme du correcteur numérique correspondant

**Objectifs**

- **Identifier** la structure d'un système asservi.
- **Proposer** une démarche de réglage d'un correcteur P, PI , PID.
- **Justifier** le choix d'un correcteur
- **Analyser** et **interpréter** un algorithme.
- **Réaliser** une dérivée et une intégration sous forme numérique
- **Générer** un programme et l'implanter dans un système cible
- **Effectuer** des manipulations sur un système afin d'identifier son comportement

**Activité 1**

Réglage expérimental des paramètres d'un correcteur

**Activité 2**

Mise en œuvre d'un algorithme pour modéliser un correcteur numérique

**Activité 3**

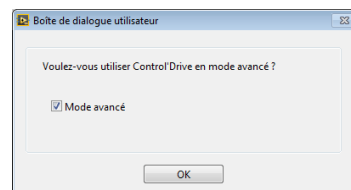
Activité séparée sur Robot Nao

**Déroulement**

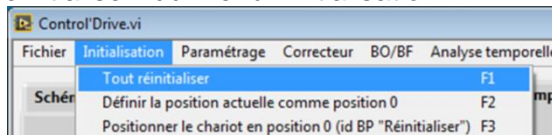
On cherche à mettre en place une démarche permettant d'optimiser les performances de positionnement de l'axe.

Mettre en fonctionnement Control'X : Basculer l'interrupteur général au dos de Control'X, fermer le capot en plexiglas et armer le système. Toutes les diodes doivent s'allumer.

☞ Lancer maintenant le logiciel **Control'Drive en mode avancé**  
Sauf s'il ne l'est déjà, le chariot de Control'X doit s'initialiser à gauche.



☞ Si Control'Drive a été lancé avant d'armer Control'X, effectuer une **réinitialisation** en utilisant la fonction "Tout réinitialiser" du menu "Initialisation" :



☞ Vérifier que le pilotage se fait en **boucle fermée** (Menu "BO/BF")

**Localisation des éléments de l'asservissement**

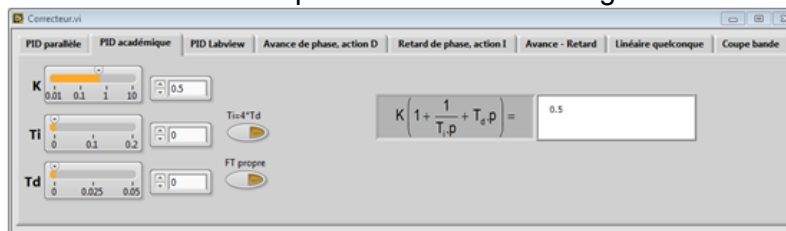
**D1.** A partir des éléments donnés dans le Dossier Technique, dessiner le schéma bloc simplifié de l'asservissement en repérant les divers éléments concernés.

**D2.** Déterminer les paramètres à modifier pour intervenir sur les actions P I et D.

**D3.** Rappeler les 3 critères de qualité d'un asservissement

**Modification des paramètres du correcteur**

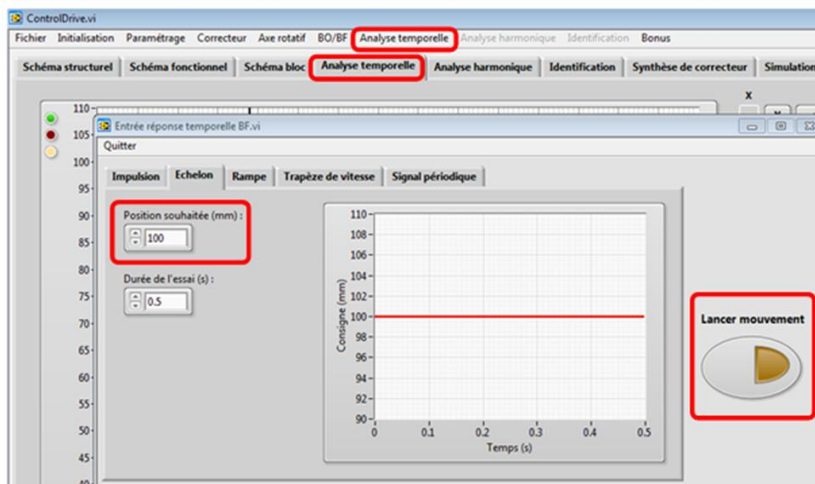
☞ Le réglage du correcteur PID se fait par l'intermédiaire du logiciel dans le menu "correcteur"



Les valeurs des coefficients **K, Ti et Td** correspondent au correcteur suivant:

$$C(p) = K(1 + \frac{1}{T_i \cdot p} + T_d \cdot p)$$

☞ On travaille ici dans l'onglet "Analyse temporelle". Générer une consigne en échelon de 100 mm d'amplitude : menu "Analyse temporelle", "Définir entrée".



**D4. Essais en mode proportionnel P :**

Lancer une mesure avec une consigne de 100mm ° pour différentes valeurs de **K (0-0,1-0.3-0,5-1-3)** Conclure sur la précision et la stabilité.

**D5. Essais en mode proportionnel intégral PI :**

Lancer une mesure avec une consigne de 100mm avec **K =0.5** et **Ti variable**. Conclure sur la précision et la stabilité.

**D6. Essais en mode proportionnel intégral dérivé PID :**

Lancer une mesure avec une consigne de 100mm avec pour **K = 0.5**, **Ti et Td** étant à choisir. Conclure sur la précision et la stabilité.

**D7.** Conclure sur les effets du réglage du correcteur : sont-ils bien ceux attendus conformément à la théorie...?

**Essai en limite d'instabilité ou « pompage », réglage de Ziegler et Nichols)**

Une façon pratique de régler un asservissement, a été donnée par Ziegler et Nichols dans les années 50.

Le correcteur PID a alors pour équation :  $C(p) = K_r \cdot (1 + \frac{1}{\tau_i \cdot p} + \tau_d \cdot p)$

En boucle fermée, on augmente progressivement la valeur de la correction proportionnelle seule, les autres coefficients étant à zéro.

On relève la valeur de  $K_r$  limite, notée  $K_o$ , qui donne une situation instable pour l'asservissement (oscillation), ainsi que la période de cette oscillation notée  $T_o$ .

Il suffit alors de régler les coefficients du PID selon le tableau suivant :

Type de correcteur	Gain $K_r$	$\tau_i$	$\tau_d$
Proportionnel	$0,5.K_o$		
PI	$0,45.K_o$	$0,83.T_o$	
PID	$0,6.K_o$	$0,5.T_o$	$0,125T_o$

**D8.** Suivre cette méthode pour un déplacement de 100mm et relever l'allure de la réponse après avoir correctement réglé votre correcteur PID.

**D9.** Conclure sur ce réglage. Comparer aux courbes obtenues sur le système réel avec ses valeurs par défaut.

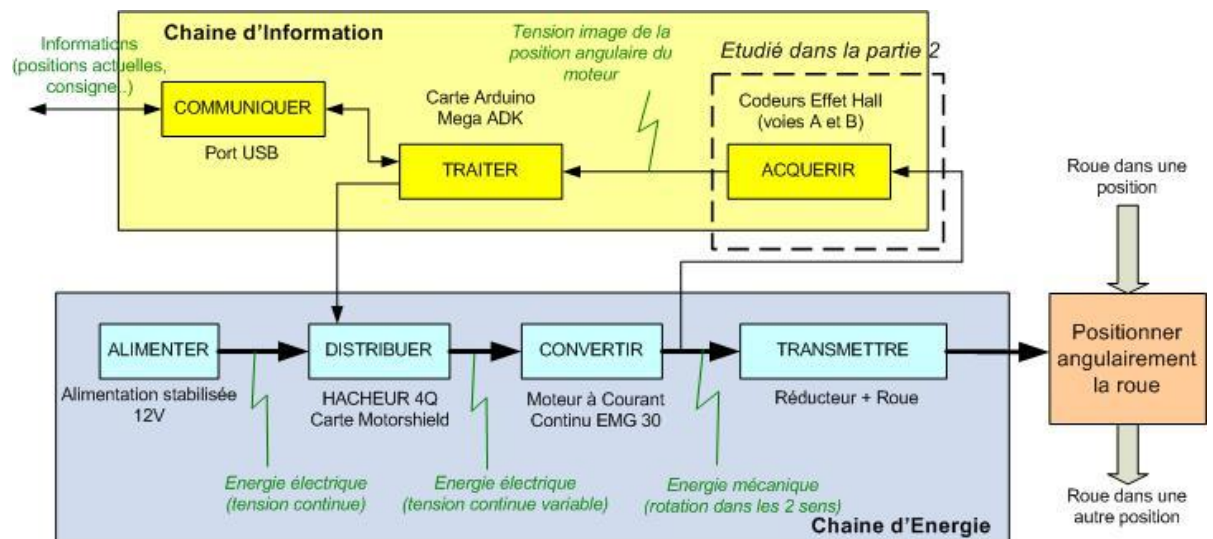
**Documents / Matériel**

- un poste informatique équipé du logiciel « Arduino »
- une carte « Arduino méga ADK »
- un ensemble MCC / réducteur / codeurs type « EMG30 » / charge
- Une alimentation stabilisée réglable
- Une boîte de masses étalonnées

**Déroulement**

Informations pratiques

Se connecter au PC en mode « Elève » → pas de mot de passe ! Les fichiers dont vous avez besoin se situent dans le répertoire « ATS » sur le bureau du PC



Dans cette activité, nous allons nous intéresser à différentes stratégies d'asservissement numérique en position d'un système poulie-câble permettant de soulever une masse (sous-ensemble du système).

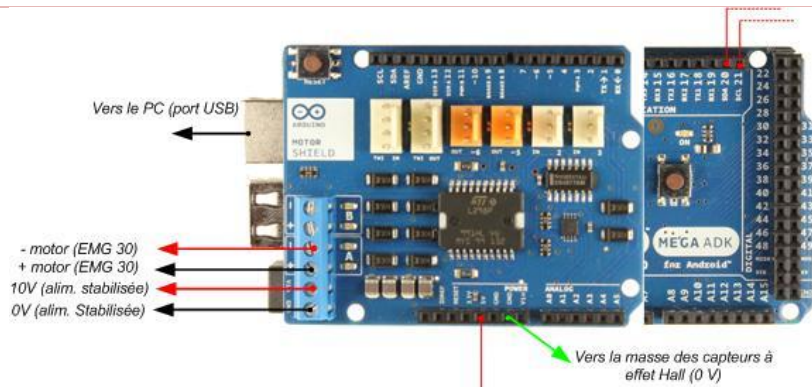
L'ensemble EMG30 est composé d'une MCC, d'un réducteur, de 2 codeurs et d'une roue. Des informations techniques sont fournies dans le fichier « EMG 30.pdf ».

La carte Arduino communique en « temps réel » des informations grâce à une communication série (via le port USB).

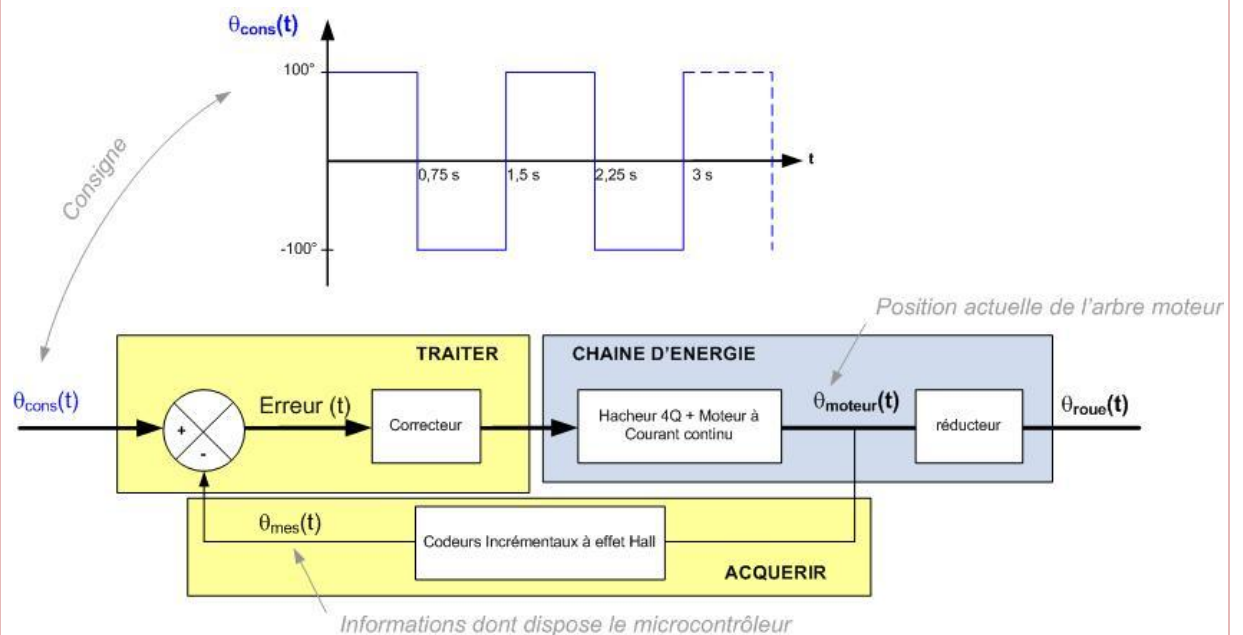
Le module « Encoder.h » importé permet de traiter automatiquement les signaux issus des codeurs et de calculer une position mémorisée dans la variable « positionActuelle ».

**Rappel sur les codeurs optiques de position :**

- D10.** Tracer l'allure des tensions mesurées UA et UB en sortie du montage (préciser abscisses, ordonnées et valeurs caractéristiques).
- D11.** Quel traitement numérique devra réaliser le microcontrôleur « arduino » pour connaître en temps réel la position angulaire de la roue ?
- D12.** Quel traitement de signal permet au microcontrôleur de connaître le sens de rotation du moteur et donc d'incrémenter ou décrémente le compteur ?
- D13.** Quelle est la plus petite position angulaire (en degrés) de la roue mesurable (encore appelée résolution de mesure de la position angulaire de la roue) ? En déduire la plus petite position angulaire de l'arbre moteur mesurable (en degrés).



Nous souhaitons piloter le moteur en position angulaire et imposer une consigne particulière que nous noterons  $\theta_{\text{cons}}(t)$  (consigne de position angulaire de la roue). Cette consigne va nous permettre de comparer les performances de l'asservissement de position réalisé. Pour cela, nous avons besoin d'une structure de type **asservissement** : on compare (en réalisant une soustraction) la position à atteindre (la consigne :  $\theta_{\text{cons}}$ ) avec la position actuelle mesurée (cf. figure ci-dessous).



Le moteur à courant continu est piloté par un hacheur 4 quadrants intégré à une carte appelée « Motorshied », qui permet de faire tourner le moteur dans les 2 sens de rotation et également de le freiner. Celle-ci est prévue pour être connectée et facilement pilotée par une carte « Arduino ».

Piloter la vitesse d'une MCC revient à faire varier sa tension d'alimentation (en faisant varier le rapport cyclique sur 8 bits). La consigne de rapport cyclique est un mot binaire codé sur 8 bits : 0 correspond à un rapport cyclique de 0 et 255 correspond à un rapport cyclique de 1.

**Si vous avez le moindre doute sur les activités qui suivent, posez votre question !**  
**Les risques sont :** - collision de la masse soulevée avec le carter de protection ;  
 - surcharge intensité moteur entraînant sa destruction.

### Procédure d'utilisation du système

- Débrancher le moteur de l'alimentation stabilisée.
- Brancher la carte Arduino Mega ADK via le port USB au PC.
- Connecter la carte Arduino.
- Mettre sous tension l'alimentation stabilisée et la régler sur 10V (ne pas brancher le moteur)
- **DEBRANCHER LE FIL** qui relie le moteur à l'alimentation stabilisée
- **TELEVERSEMENT** du programme ou **CLIC** sur moniteur série
- **POSITIONNER** la masse en bas à chaque réinitialisation du programme ;
- Vérifier que la course de la masse est correcte
- Brancher le moteur à l'alimentation stabilisée



## Commande proportionnelle :

Algorithme du correcteur Proportionnel :  $U_{moy}(t) = K * Erreur(t)$

Avec  $U_{moy}$  la tension moyenne aux bornes du moteur

$K_p$  : le coefficient du correcteur proportionnel

**D14.** Débrancher l'alimentation stabilisée de la carte puis ouvrir et téléverser le programme intitulé « asservissement.ino ». Régler le paramètre K à la valeur souhaitée ( $K_i=0$  et  $K_d=0$ ). Vérifier que la constante  $U_{lim}$  est définie à la valeur 10. On pourra commencer par une valeur de  $K=3$ .

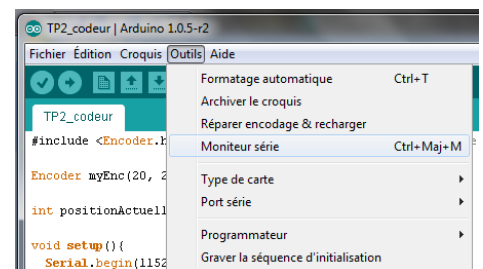
**D15.** Analyser globalement la structure du programme et plus particulièrement la fonction « correcteur ».

**Mettre en place une masse de 20g puis augmenter progressivement au fur et à mesure des questions la valeur de K et ne pas laisser fonctionner le système pendant un long temps de fonctionnement (soucis d'échauffement du moteur).**

**D16.** Connecter l'alimentation stabilisée en la réglant sur 10 V. Observer les positions angulaires de la roue au cours du temps pour des valeurs différentes de K ainsi que pour des valeurs différentes de la variable « rapportCyclique » (255 puis 128). Pour afficher les positions sous Excel : cf. fichier « exploitation\_mesures.pdf ».

Pour communiquer avec la carte « arduino » et récupérer des mesures, ouvrir le fichier « moniteur série ». Lorsque le message « fin d'initialisation » s'affiche, le microcontrôleur « arduino » exécute la boucle « loop » à l'infini.

⚠ Attention : ne pas laisser le moniteur mémoriser des données pendant un temps trop important (temps fonctionnement < 5 min sinon risque de plantage du PC).



**D17.** Observer les positions angulaires de la roue au cours du temps pour des valeurs différentes de « K » en suivant la procédure d'utilisation du système (débrancher l'alim avant chaque téléversement) et trouver une valeur de K qui ne donne pas lieu à des dépassements de la consigne (100° ou -100°).

**D18.** Trouver une valeur de K qui donne lieu à des dépassements de consigne mais dont la position angulaire se stabilise au cours du temps (système stable).

**D19.** Conclure vis à vis des résultats obtenus.

## Principe d'une commande PID

Nous voulons pouvoir implémenter une commande Proportionnelle Intégrale Dérivée **numérique** comme dans le système réel que vous étudiez.

Nous souhaitons approcher au mieux la fonction du régulateur PID analogique suivant

$$U(t) = K \cdot \left( erreur(t) + \frac{1}{T_i} \cdot \int_0^t erreur(t) dt + T_d \cdot \frac{d}{dt} erreur(t) \right)$$
$$erreur(t) = \theta_{cons}(t) - \theta_{mes}(t)$$

Dans la mesure où la période d'échantillonnage  $T_e$  est suffisamment petite, on peut se contenter de remplacer l'équation différentielle du régulateur PID par une équation aux

différences prenant en compte les équivalents numériques des écarts aux instants d'échantillonnage.

Vous trouverez (en annexe) une documentation sur la correction numérique des systèmes.

**D20.** Prendre connaissance de l'annexe sur les algorithmes des correcteurs numériques PID. Retrouver en dessinant un exemple de courbe les relations numériques correspondant aux actions intégrale et dérivée.

**D21.** Retrouver les lignes de programme correspondant à ces paramètres.

### **Commande Proportionnelle Intégrale**

**Pensez bien à débrancher l'alim avant chaque téléversement !**

**D22.** Toujours avec la masse de 20g, observer les réactions du système pour des couples de valeurs  $K/K_i$  variées et déterminer des valeurs pour lesquelles on n'observe pas d'oscillations et d'autres pour lesquelles il y en a. Vous tracerez dans chaque cas les courbes obtenues.

**D23.** Pour chaque essai retenu : déterminer le temps de réponse à 5% (faire apparaître clairement le tracé), l'erreur statique, la valeur du dépassement (en % de la consigne)

**D24.** Choisissez un couple de valeurs  $K/K_i$  qui vous paraît optimal, puis modifier la masse  $m$  et refaire les essais.

**D25.** Conclure sur un réglage optimal de l'asservissement numérique.