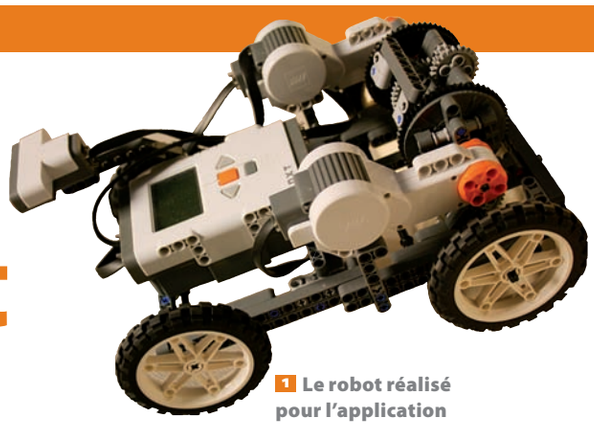
Les Lego NXT se positionnent



1 Le robot réalisé pour l'application

FRANÇOIS LOUF, CLÉMENT ROCH, HECTOR LETELLIER, JEAN-LAURENT DUCHAUD, LAURENT CHAMPANEY [1]

Voici un exemple d'utilisation des Lego NXT pour la formation aux sciences de l'ingénieur. L'objectif de ce travail est de montrer comment on peut, à partir de ces boîtes de jeux, introduire différentes notions du domaine de l'asservissement.

© LEGO

2 La boîte Mindstorms éducation et son automate programmable

Les objectifs et les moyens à disposition

Les travaux présentés ici ont été, pour partie, réalisés par et pour deux élèves de troisième, en stage au département de génie mécanique de l'ENS de Cachan. Le côté ludique et les facilités d'utilisation de leur support permettent d'en envisager des usages au niveau d'une classe de seconde. En effet, le robot réalisé **1** utilise une boîte de Lego Mindstorms éducation (réf. 9797) **2** et une extension (réf. 9648). Le schéma cinématique de la transmission est présenté en **3**.

Dans un premier temps, nous avons choisi de n'alimenter et de ne piloter qu'un seul des deux moteurs. Nous détaillerons l'écriture, puis le test, d'un programme permettant d'asservir en position le mobile grâce à un correcteur proportionnel, et les résultats expérimentaux mettront en évidence les limites de ce type de correction.

Une modification de ce programme va ensuite y introduire la prise en compte du cumul de l'erreur au cours du temps. Pour cela, nous avons utilisé une action intégrale sur l'alimentation du second moteur. Les expériences réalisées permettent de constater de nettes améliorations par rapport au cas précédent, mais aussi des limites.

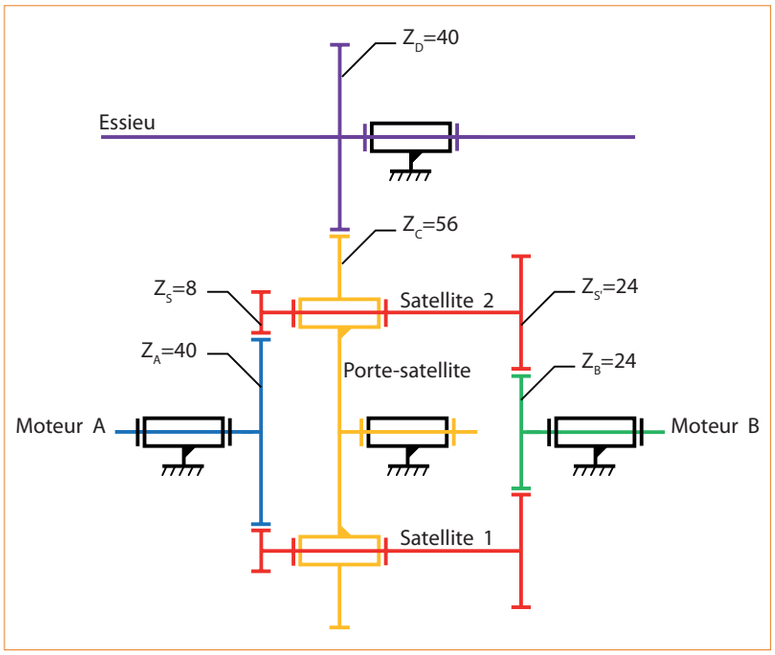
Pour mettre en évidence le rôle de chaque paramètre, il est nécessaire de modifier les gains introduits directement dans le programme. Afin de rendre les expérimentations plus aisées, nous proposons, pour finir, des solutions permettant de régler facilement les deux gains.

mots-clés automatismes, dynamique, travaux pratiques

L'architecture du robot

Le robot peut se déplacer en avançant et en reculant. Il est réalisé à partir d'un châssis supportant la motorisation, la transmission mécanique, l'essieu avant non moteur, l'essieu arrière moteur ainsi que l'automate programmable de commande des moteurs et le capteur de mesure de la position du véhicule.

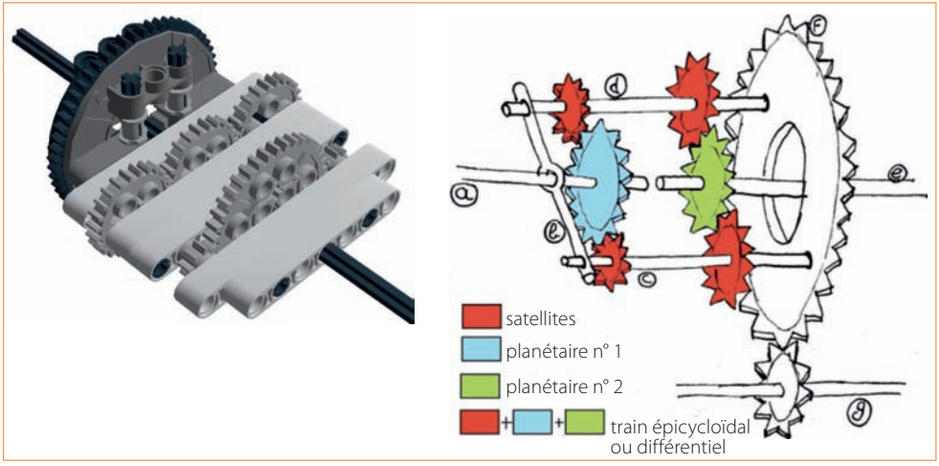
Nous avons choisi de réaliser la transmission mécanique, liant les moteurs aux roues motrices, par un train épicycloïdal de type III **4**. Cela va permettre de montrer la complémentarité des actions des correcteurs proportionnel et intégral, qui pourront être injectées de manière indépendante dans deux actionneurs différents. L'objectif est notamment de mettre en évidence l'intérêt de la correction intégrale pour corriger une erreur statique due aux frottements, par exemple.



3 Le schéma cinématique de la transmission mécanique

[1] Respectivement : maître de conférences ; élèves de troisième en stage d'observation ; élève normalien ; professeur des universités. Contact : francois.louf@dgm.ens-cachan.fr.

[2] Les chiffres gris entre crochets renvoient aux références en encadré



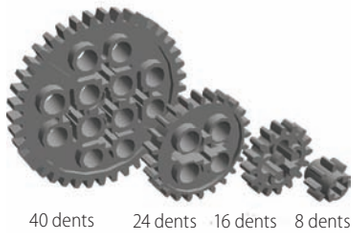
4 Le train épicycloïdal en image CAO et en dessin réalisé par un élève de troisième

Les contraintes de réalisation du train épicycloïdal sont :

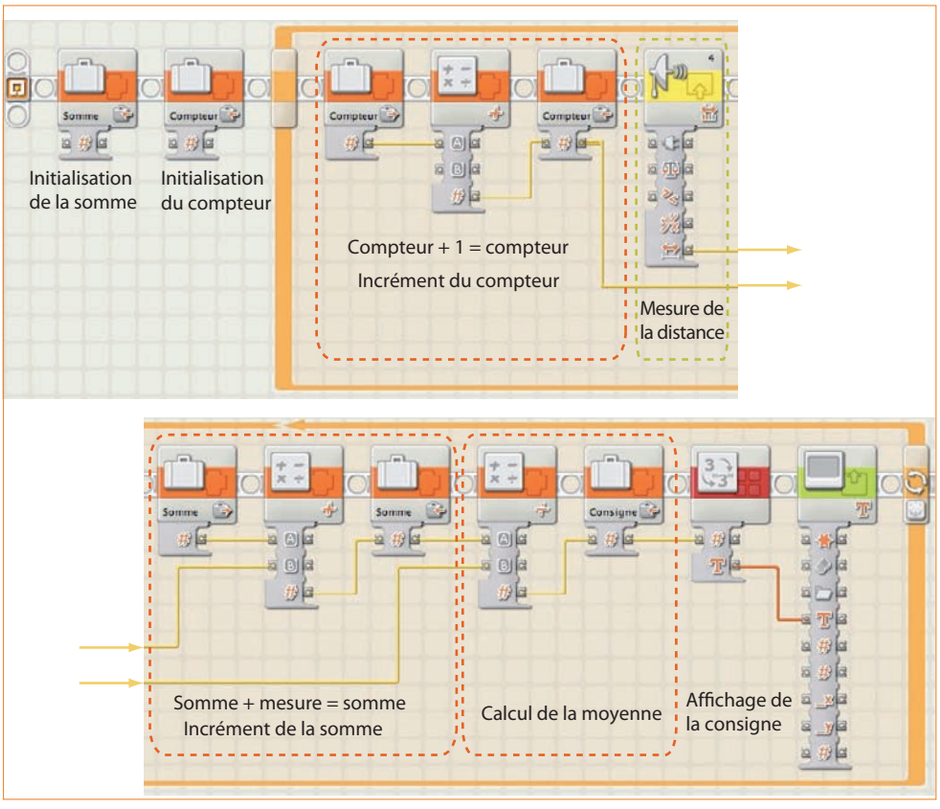
- Posséder deux planétaires à denture extérieure. En effet, il n'existe pas de couronne intérieure de diamètre important.
- Quatre tailles de pignons sont disponibles **5**.
- L'entraxe entre deux pignons, si l'on réalise une construction classique, ne peut varier que de façon discrète et est lié à une unité dimensionnelle de Lego ; le nombre de combinaisons possibles conduisant à un engrenage montable sur un support est donc limité.

La mesure de la distance

Pour mesurer la position du robot relativement à un obstacle, on utilisera un capteur de distance à ultrasons. Le principe de ce type de capteurs est simple : un émetteur et un récepteur sont situés dans le même boîtier ; l'émetteur envoie des ondes qui vont se réfléchir sur l'objet à détecter et ensuite revenir à la source. Le temps mis pour parcourir un aller-retour permet de calculer la distance de l'objet par rapport à l'émetteur. Mais, au-delà d'une certaine distance, les ondes se perdent ; on ne peut utiliser le capteur que pour mesurer une distance comprise dans l'intervalle [6, 250] cm environ.



5 Les 4 types de pignons disponibles



6 Le programme de calcul de la consigne

L'implémentation d'un correcteur proportionnel

Les étapes de la programmation

Pour construire le programme complet, on procède pas à pas : dans un premier temps, on réalise un élément de programme permettant de calculer la consigne (distance initiale entre le robot et un obstacle), puis un programme laissant le temps à l'utilisateur de changer cette distance par déplacement de l'un ou l'autre des deux éléments ; enfin, on réalise un programme permettant au robot de revenir, de façon autonome, à une distance de l'obstacle à peu près égale à la distance initiale.

La mesure de la distance initiale (consigne)

Le robot est posé face à un obstacle à une distance qui peut être mesurée par le capteur à ultrasons, c'est-à-dire comprise dans l'intervalle [6, 250] cm. De manière à s'affranchir d'éventuelles perturbations (vibrations), pendant 5 s, le robot fait la moyenne des distances mesurées, puis affiche la consigne à l'écran.

Le programme réalisant ce calcul est présenté en **6** .

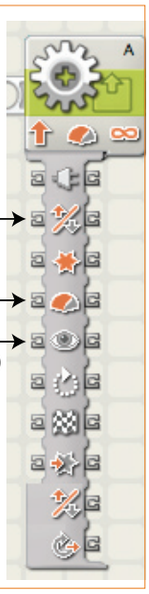
Le temps pour le déplacement du robot

Pendant 5 s, l'utilisateur doit pouvoir modifier la position du robot sans que celui-ci ne réagisse.

Le retour en position initiale

Tant qu'on laisse le programme actif, le robot doit maintenir une distance à l'obstacle égale à la consigne. Le principe est simple :

- Si le robot est trop près de l'obstacle, il faut le faire reculer : le moteur doit tourner dans le sens négatif.



7 La description partielle du « bloc moteur »

● Si le robot est trop loin de l'obstacle, il faut le faire avancer : le moteur doit tourner dans le sens positif.

Maintenant que le sens de rotation du moteur est connu, la question est de savoir à quelle vitesse le robot doit évoluer. Le « bloc moteur » 7 peut contrôler le sens de rotation, alimenter plus ou moins le moteur, et éventuellement contrôler sa vitesse.

Intuitivement, il paraît raisonnable d'aller vite lorsque l'écart est important et lentement lorsque l'écart est faible. La vitesse du véhicule est directement liée à la vitesse de rotation du moteur A, qui, elle, est imposée par la tension d'alimentation. On définit donc un correcteur proportionnel liant l'écart mesuré au régime moteur souhaité :

$$P\,K_p = K_p\,\varepsilon$$

L'algorithme à implémenter à l'aide du logiciel fourni est présenté en 8 ; la traduction de la partie interne de la boucle de cet algorithme en « langage Lego », en 9.

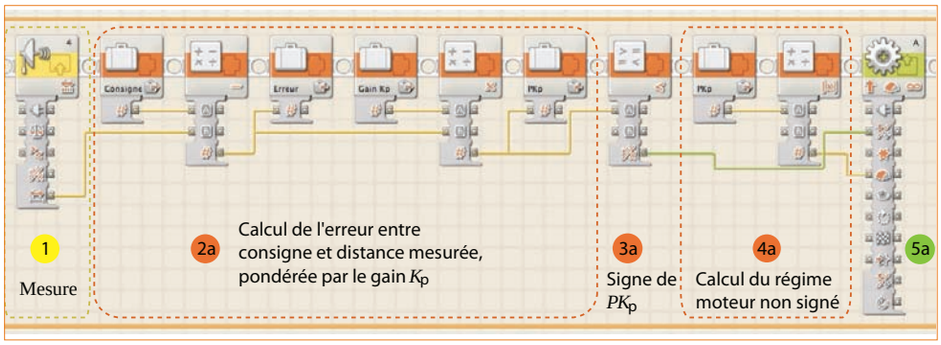
Pour donner une valeur raisonnable au gain K_p, on peut par exemple demander au moteur A de tourner à son régime maximal (indice 100) lorsque l'erreur ε (écart entre la consigne et la distance mesurée entre le robot et l'obstacle) est de 100 cm. Cela conduit donc à un K_p unitaire.

L'analyse des résultats

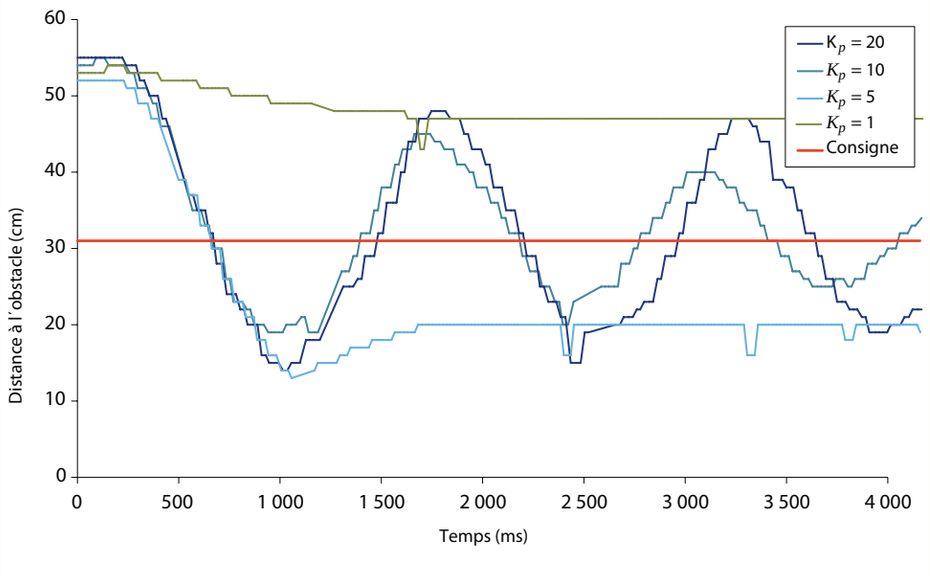
Analysons les premiers résultats expérimentaux obtenus 10 :

```
Initialiser le gain Kp                                                    0
Tant que le robot n'est pas arrêté manuellement
    Mesurer la distance courante : D                                      1
    Calculer l'écart à la consigne C : ε = C – D                          2a
    Calculer le signe de l'écart                                          3a
    Calculer la consigne moteur non signée : P Kp = |Kp ε|                4a
    Commander le moteur A avec le bon sens de rotation et le bon niveau   5a
Fin « tant que »
```

8 L'algorithme de l'asservissement en position avec un correcteur proportionnel



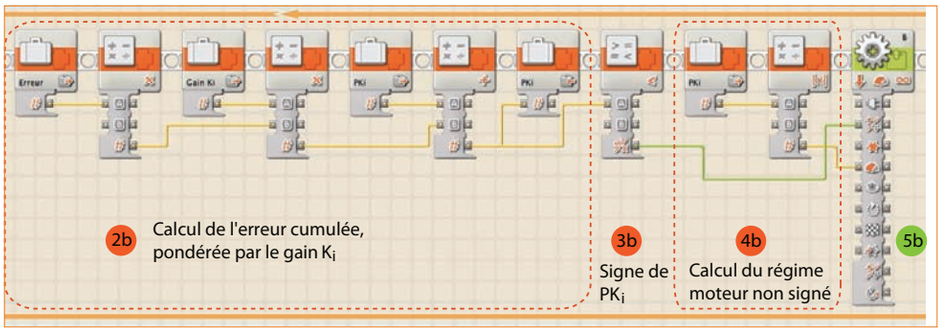
9 Le programme Lego réalisant l'asservissement en position avec un correcteur proportionnel



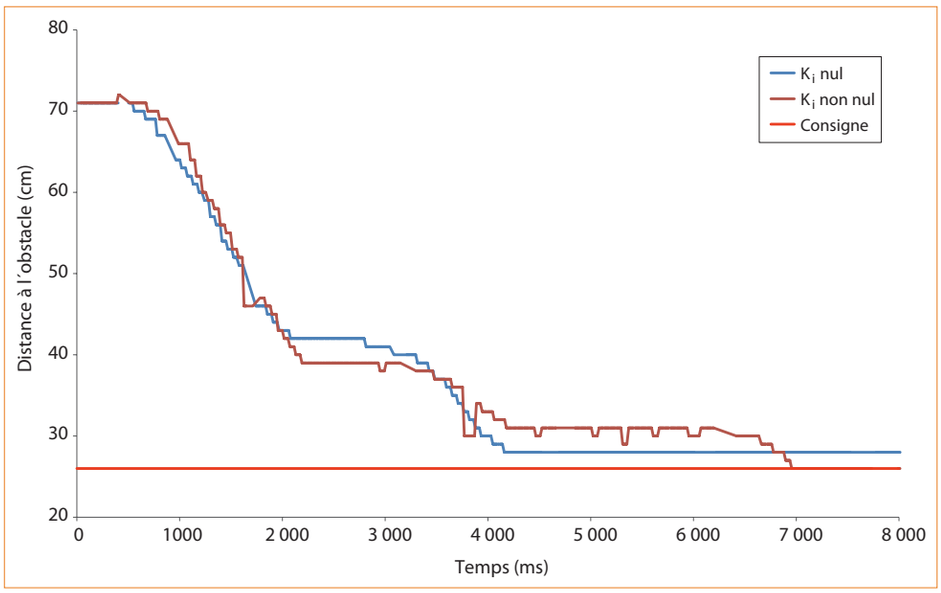
10 L'évolution de la distance à l'obstacle au cours du temps pour différents réglages de K_p

```
Initialiser le gain Kp et le gain Ki                                      0,1
Initialiser la somme des erreurs P Ki                                     0,2
Donner une valeur au temps nécessaire pour faire une boucle : Δt = 10 ms  0,3
Tant que le robot n'est pas arrêté manuellement
    Mesurer la distance courante : D                                      1
    Calculer l'écart à la consigne C : ε = C – D                          2a
    Calculer le signe de l'écart                                          3a
    Calculer la consigne moteur non signée : P Kp = |Kp ε|                4a
    Commander le moteur A avec le bon sens de rotation et le bon niveau   5a
    Mettre à jour l'erreur cumulée : P Ki = Ki ε Δt + P Ki                2b
    Calculer le signe de la somme                                         3b
    Calculer la consigne moteur non signée : |P Ki|                       4b
    Commander le moteur B avec le bon sens de rotation et le bon niveau   5b
Fin « tant que »
```

11 L'algorithme de l'asservissement avec un correcteur proportionnel/intégral



12 Les éléments supplémentaires à intégrer dans le boucle de la figure 9 permettant de réaliser l'asservissement en position avec un correcteur proportionnel/intégral



13 L'évolution de la distance à l'obstacle au cours du temps pour différents réglages de K_i

- Le temps mis par le robot à retrouver approximativement sa position initiale dépend du gain K_p : plus le gain est grand, plus le robot est réactif.
- Si le gain est faible, le robot peut ne jamais revenir en position : l'écart permanent (erreur statique) entre la position initiale et la position courante est important.
- Si le gain est trop grand, le comportement du robot est oscillant, voire même instable.

Les frottements dans la chaîne cinématique et la résistance au roulement des roues sur le sol sont tels qu'une alimentation trop faible des moteurs ne permet pas de faire avancer le mobile. Celui-ci demeure donc à l'arrêt avec une erreur permanente. Ce point montre qu'il est nécessaire d'apporter un élément supplémentaire dans la correction : le fait qu'on cumule une erreur, même faible, pendant très longtemps doit permettre de modifier la position au bout d'un certain temps.

Le fait que le signal temporel soit crénelé s'explique par la résolution du capteur à ultrasons, qui est de l'ordre du centimètre. Ainsi, si le mobile avance à une vitesse faible, il faut un grand nombre de millisecondes avant que la quantité issue du capteur ne s'incrémente de 1 cm.

L'implémentation d'un correcteur proportionnel/intégral

Les étapes de la programmation

Les mesures précédemment réalisées ont mis en évidence que le principe de correction utilisé pour le moment n'était pas forcément suffisant : une erreur permanente existe, faible ou grande dans certains cas. On propose donc d'introduire un élément dans le programme permettant de cumuler cette erreur au cours du temps, d'alimenter le moteur A comme précédemment, et le moteur B par une quantité image de cette erreur cumulée.

Les modifications à apporter au précédent algorithme sont présentées en 11. On voit que la première partie de la boucle est inchangée. L'action intégrale est associée uniquement au moteur B. La traduction de cet algorithme en langage Lego est présentée, pour la partie interne à la boucle, en 12.

Les résultats

Les résultats expérimentaux 13 montrent tout d'abord que, dans le cas précédent où K_p était faible, l'action intégrale permet, au bout d'un certain temps dépendant du gain, qu'un K_i nul conduise à une erreur permanente, et qu'un K_i non nul annule cette erreur. On peut donc grâce au correcteur K_i vaincre les frottements et modifier la position du robot. Les résultats obtenus par les élèves mettent aussi en évidence que K_i entraîne des oscillations dans le comportement, voire une instabilité.

Remarque

Le choix fait ici d'utiliser deux motorisations indépendantes accouplées via un train épicycloïdal est original, car il permet d'appliquer indépendamment les actions proportionnelle et intégrale sur deux actionneurs. Toutefois, le comportement de l'ensemble est plus difficile à prévoir, le choix des paramètres délicat. Pour plus de simplicité, tout ce qui a été expliqué ici peut être transposé à un mobile actionné par un ou plusieurs moteurs associés à une chaîne cinématique à une seule mobilité, c'est-à-dire montés sur un même arbre de transmission. On obtient alors des courbes de comportement très classiques 14 :

- On visualise parfois une erreur statique très importante.
- On visualise parfois un comportement instable.
- On observe des comportements oscillants amortis de forte amplitude.
- On parvient à trouver un compromis avec un faible dépassement, une réponse rapide, et une erreur statique nulle.

Étude mécanique de transmission

Étude cinématique du système

La relation cinématique liant la vitesse de rotation des moteurs A et B à la vitesse du porte-satellite est donnée par la formule de Willis :

$$\omega_A - \lambda\omega_B + (\lambda - 1)\omega_C = 0$$

λ : raison basique du train

Ici, la raison basique vaut :

$$\lambda = (8 / 40)\ (24 / 24) = 1 / 5$$

La vitesse de l'essieu est donc définie par la vitesse de rotation des moteurs A et B selon la relation :

$$\omega_D = -\ (Z_C / Z_D)\ \omega_C = (Z_C / Z_D)\ [(\lambda\omega_B - \omega_A) / (1 - \lambda)]$$

Avec l'architecture retenue, l'application numérique donne la relation suivante :

$$\omega_D = (7 / 4)\ (0{,}2\ \omega_B - \omega_A)$$

Ainsi, si on alimente uniquement le moteur A, et que le moteur B est bloqué, on aura :

$$\omega_D = -\ (7 / 4)\ \omega_A$$

Si on alimente uniquement le moteur B, et que le moteur A est bloqué, on aura :

$$\omega_D = (7 / 20)\ \omega_B$$

La vitesse maximale est obtenue lorsque les deux moteurs tournent à plein régime, mais en sens opposés :

$$\omega_{D_{max}} = 2{,}1\omega_{mot_{max}}$$

Le rayon *r* des roues étant de 4,1 cm, et la vitesse de rotation maximale à vide des moteurs $\omega_{mot_{max}}$ étant de 170 tr·min^{-1}, soit 17,8 rad·s^{-1}, la vitesse maximale atteinte par le véhicule est :

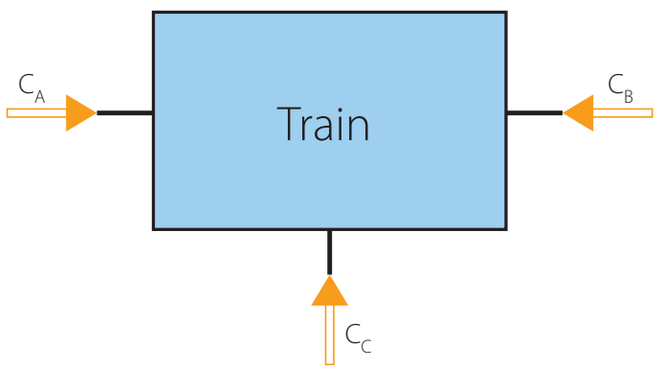
$$V_{max} = r\omega_{D_{max}} = 1{,}53\ \text{m}\cdot\text{s}^{-1}$$

Étude dynamique du système complet

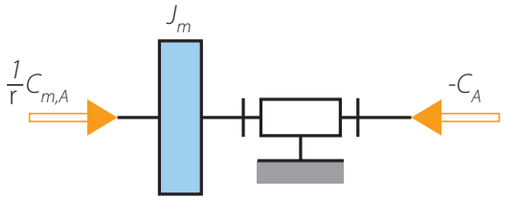
Dans cette partie, on cherche la réponse temporelle du système complet lorsqu'on alimente les moteurs A et B par deux tensions U_A et U_B.

On fait l'hypothèse que les inerties des pièces constituant le train sont négligeables et que les liaisons sont sans frottement. On isole le train et on note a1 :

C_A le couple appliqué sur l'arbre A du train
C_B le couple appliqué sur l'arbre B du train
C_C le couple appliqué sur l'arbre C du train



❶ La convention utilisée pour le signe des couples



❷ Le bilan des couples sur l'arbre moteur

a Les couples appliqués sur les planétaires A et B et le porte-satellite C du train épicycloïdal et les actions mécaniques extérieures sollicitant l'arbre de sortie du motoréducteur A

Le théorème de l'énergie cinétique fournit alors les relations classiques liant les couples :

$$C_A = C_C\,[1 / (\lambda - 1) \qquad \text{et} \qquad C_B = -\,C_C\,[(\lambda / (\lambda - 1)]$$

En isolant l'arbre moteur A a2, on a l'équation dynamique suivante :

$$J_m\,\dot{\omega}_A = (1 / r)\,C_{m,A} - C_A$$

J_m : inertie du rotor et du train d'engrenages ramenée sur l'arbre de sortie du servomoteur (en aval du réducteur interne)
$C_{m,A}$: couple moteur appliqué sur le rotor
$r < 1$: rapport de réduction du réducteur interne au servomoteur Lego

On a bien entendu la même relation pour l'arbre moteur B, identique en termes d'architecture :

$$J_m\,\dot{\omega}_B = (1 / r)\,C_{m,B} - C_B$$

En isolant l'arbre de sortie C, en négligeant tout effort résistant tel que les frottements, en supposant que le mobile évolue à plat, et en écrivant le principe fondamental de la dynamique, on obtient une troisième équation dynamique :

$$J_{eq}\,\dot{\omega}_C = -\,C_C$$

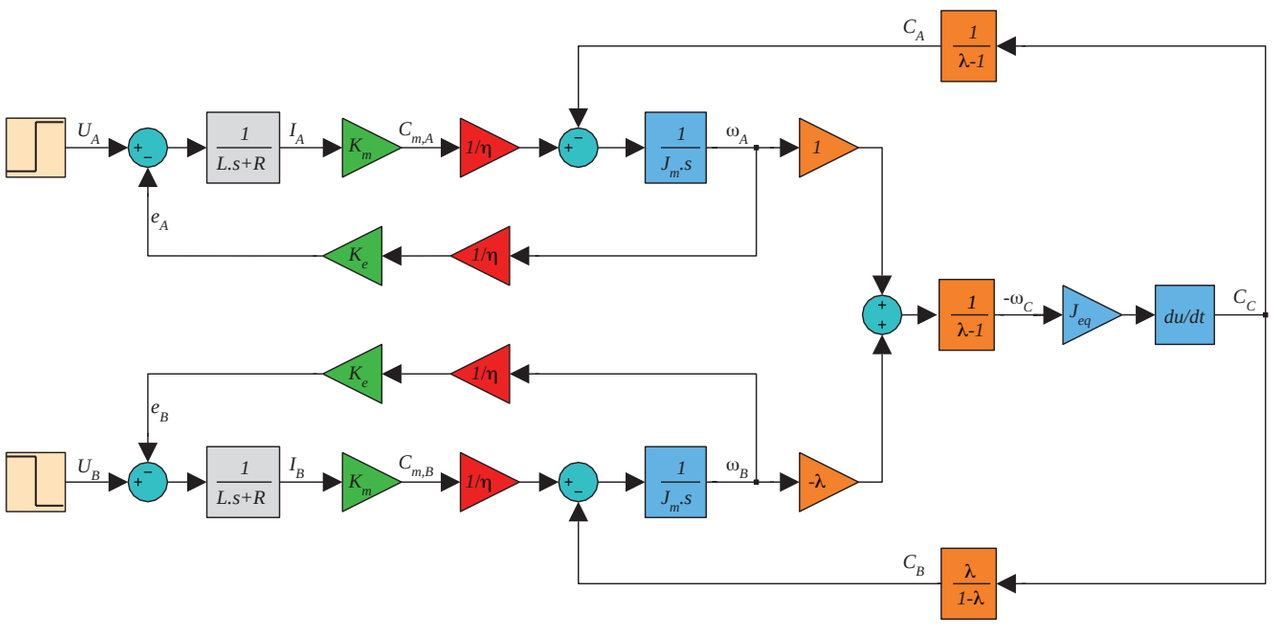
J_{eq} représente l'inertie équivalente au mobile ramenée sur l'arbre C. Ici, on ne tient compte que de la masse M du véhicule :

$$J_{eq} = M\,[(Z_C / Z_D)\,r]^2$$

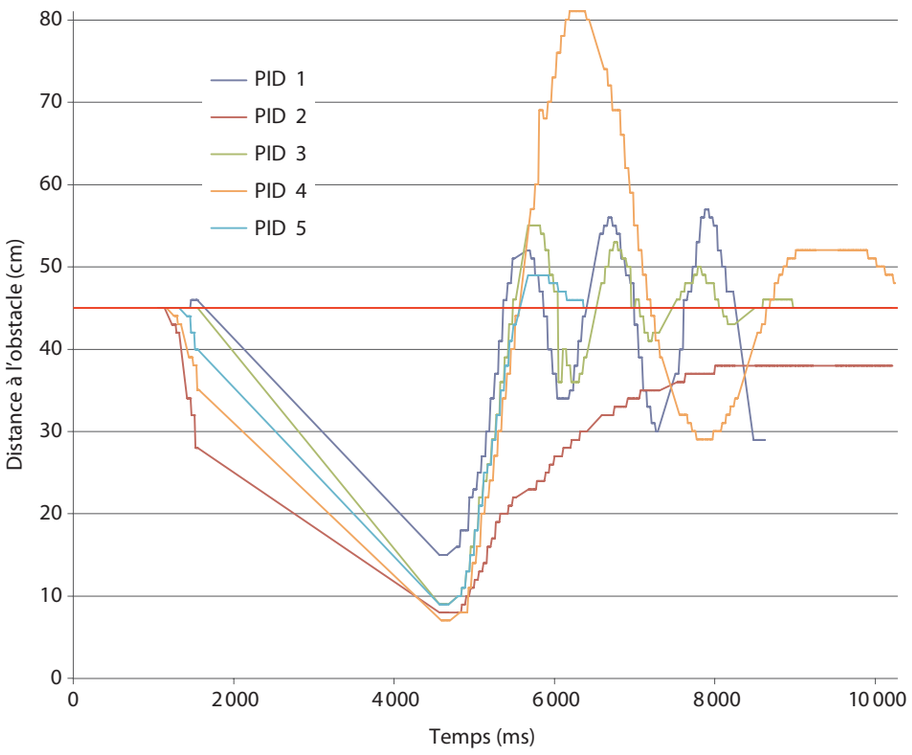
Il nous reste à écrire les équations du moteur à courant continu. Ces équations sont de la même forme pour les moteurs A et B :

$$U_{A,B} = e_{A,B} + Ri_{A,B} + L(di_{A,B} / dt)$$
$$C_{m,A,B} = Ki_{A,B}$$
$$e_{A,B} = K\omega_{A,B}$$

On a donc un problème à 12 équations (1 relation cinématique, 2 relations liant C_A, C_B et C_C, 3 relations dynamiques, 6 équations moteur) et 12 inconnues (3 vitesses angulaires, 3 couples d'entrée du train, 2 couples moteur, 2 tensions, 2 intensités). Les données du problème sont les tensions d'alimentation U_A et U_B. Le problème, qui prend la forme d'un système d'équations différentielles, peut donc être résolu. Le schéma b traduit ce problème de mécanique sous la forme d'un schéma-bloc du type de celui qu'on pourrait construire dans Simulink, par exemple, pour simuler le comportement du système en boucle ouverte ou fermée.



b Le schéma-bloc représentant le système d'équations différentielles qui régit le comportement dynamique de l'ensemble lorsque les deux moteurs sont alimentés par des tensions UA et UB



14 Les résultats expérimentaux obtenus pour un arbre moteur unique

La modification *in situ* des paramètres K_p et K_i

En pratique, il est assez laborieux de modifier les gains K_p et K_i dans le programme, de compiler celui-ci, puis de charger le résultat sur la brique pour chaque jeu de paramètres. Pour remédier à ce problème, deux solutions sont possibles :

- Modifier au début de l'exécution du programme les valeurs K_p et K_i à l'aide des boutons présents sur l'automate.
- Utiliser deux potentiomètres externes, reliés à la brique, pour modifier *on line* les valeurs des deux gains K_p et K_i.

C'est la seconde solution qui a été retenue, comme dans **[1]** [2], car elle permet d'ouvrir des perspectives : la création de capteurs adaptés aux Lego NXT. Nous allons détailler l'interfaçage de ces potentiomètres avec la brique ainsi que le programme permettant de modifier les gains à partir de ces réglages.

Le principe du capteur potentiométrique

On utilise ici des potentiomètres classiques très compacts **15**, dont on fait varier la résistance à l'aide d'un petit tournevis plat. Les potentiomètres utilisés ont une résistance maximale de $R_{max} = 20\ k\Omega$, et la résistance varie sur $N_{max} = 22$ tours.

On suppose que la variation de résistance est linéaire. Si on note $R(n)$ cette résistance, et n le nombre de tours effectués, on a donc :

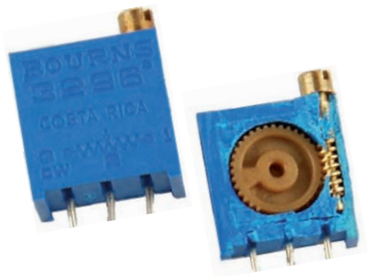
$$R(n) = (R_{max} / n_{max})\ n$$

La mesure de $R(n)$ permet donc d'avoir une information sur n : c'est le principe des capteurs potentiométriques, que l'on retrouve sur bon nombre de systèmes dans nos salles de travaux pratiques.

L'interfaçage avec la brique

Chaque câble fourni avec la brique contient six fils, un blanc, un noir, un rouge, un vert, un jaune et un bleu. Pour cette application, on coupera un câble en deux, dont on dénudera le fil blanc et le fil noir. Le branchement de ces fils sur la brique et le convertisseur analogique-numérique (CAN) sont présentés en **16**. On reconnaît sur cette figure la structure d'un pont diviseur, qui permet de calculer la tension U_R aux bornes du potentiomètre en fonction de la tension d'alimentation U, de la résistance interne à la brique R_i et de la résistance variable $R(n)$:

$$U_R = [R(n) / R_i + R(n)]\ U$$



15 Le potentiomètre multitour utilisé

On note au passage que la tension maximale aux bornes du potentiomètre est telle que :

$$U_{R\,max} = [R_{max} / (R_i + R_{max})]\ U$$

La tension d'alimentation U étant de 5 V et la résistance interne R_i valant 10 kΩ, on trouve que :

$$U_{R\,max} = [20\,000 / (10\,000 + 20\,000)] \times 5 = 3{,}33\ V$$

La mesure de U_R permet de remonter à $R(n)$, et donc à n :

$$n = (R_i / R_{max})\ [U_R / (U - U_R)]\ n_{max}$$

En pratique, on n'a pas directement accès à U_R. Un convertisseur analogique-numérique transforme la tension U_R en un nombre, noté S, codé sur 10 bits. Pour une tension U_R de 5 V (cas d'une résistance variable très grande), on aurait donc une valeur en sortie du convertisseur de 1 023. En pratique, la tension maximale qui peut être atteinte par U_R dans notre cas étant 3,3 V, on aura au maximum en sortie du convertisseur :

$$S_{max} = (1\,023 / 5) \times 3{,}3 = 682$$

On a donc :

$$S = (1\,023 / U)\ U_R = 1\,023\ [R(n) / (R_i + R(n))] = 1\,023\ [R_{max}\ n / (R_i\ n_{max} + R_{max}\ n)]$$

Inverser cette relation permet de connaître le nombre de tours de potentiomètre en fonction de la sortie S du convertisseur :

$$n = [S / (1\,023 - S)]\ (R_i / R_{max})\ n_{max}$$

Dans notre cas, la connaissance du nombre de tours, de la résistance interne, donne :

$$n = 11\ S / (1\,023 - S)$$

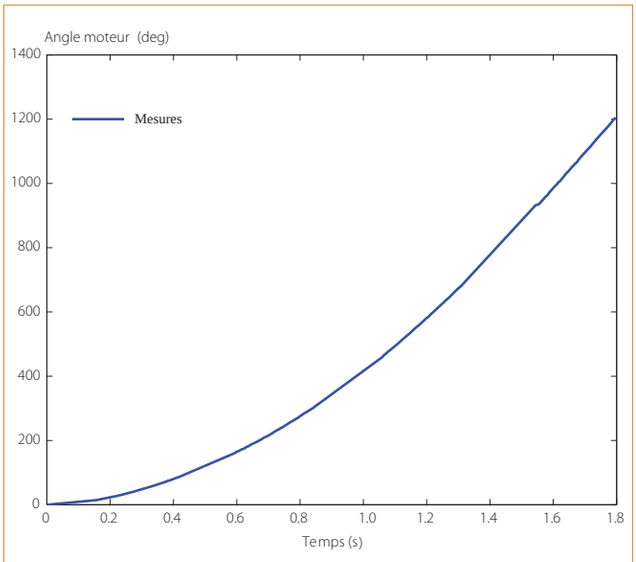
Si on souhaite faire varier linéairement K de 0 à K_{max}, il faut donc programmer la formule :

$$K = (K_{max} / n_{max})\ n \text{ soit } K = K_{max}\ (R_i / R_{max})\ [S / (1\,023 - S)]$$

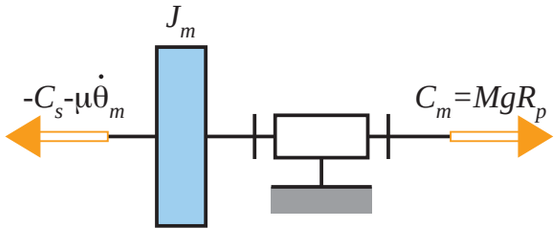
Les deux potentiomètres sont branchés sur les ports 1 et 2 de la brique selon le câblage de la figure **16**. Le premier permettra de modifier K_p, le second K_i, selon la relation précédente.

Détermination expérimentale des inerties

Les constantes nécessaires à l'élaboration de ce modèle peuvent être en partie trouvée sur la Toile [2], ou obtenues par des mesures adaptées, ce qui peut constituer un exercice pratique intéressant. Par exemple, pour obtenir un ordre de grandeur de l'inertie des différents éléments d'un servomoteur ramené sur l'arbre de sortie, on peut réaliser un essai de lâcher : une poulie est solidarisée à l'arbre de sortie ; une cordelette, au bout de laquelle est accrochée une masse, est enroulée sur la poulie. En lâchant la masse, le moteur se met à tourner, et on peut alors relever la position de l'arbre grâce au codeur dont le servomoteur est équipé. On obtient une courbe du type de celle présentée en a1. Pour estimer, à partir de cet essai, l'inertie du motoréducteur ramenée à l'arbre de sortie, un modèle de comportement doit être introduit. Ce modèle s'appuie sur le schéma proposé en a2.



❶ Résultat typique de l'essai de lâcher



❷ Modélisation de l'essai de lâcher

a La modélisation de l'essai de lâcher

Dans ce modèle, on considère que la masse génère un couple moteur sur l'arbre de sortie, via la poulie de rayon R_p :

$$C_m = M\,g\,R_p$$

On suppose que le couple résistant provient d'un frottement de type sec et d'un frottement de type visqueux :

$$C_r = -C_s - \mu\,\dot{\theta}_m$$

Le principe fondamental de la dynamique permet alors d'écrire l'équation de mouvement suivante :

$$J_m\,\ddot{\theta}_m + \mu\,\dot{\theta}_m = M\,g\,R_p - C_s$$

La solution générale de l'équation homogène associée est du type :

$$\dot{\theta}_m = A e^{-\frac{\mu}{J_m}t}$$

Une solution particulière est :

$$\dot{\theta}_m = [(M\,g\,R_p)/\mu] - C_s$$

Ainsi, la vitesse angulaire de l'arbre de sortie est donnée par :

$$\dot{\theta}_m(t) = A e^{-\frac{\mu}{J_m}t} + \frac{MgRp}{\mu} - C_s$$

Puisque la vitesse initiale est nulle (essai de lâcher), la constante peut être déterminée :

$$\dot{\theta}_m(t) = \frac{1}{\mu}(MgR_p - C_s)\left(1 - e^{-\frac{\mu}{J_m}t}\right)$$

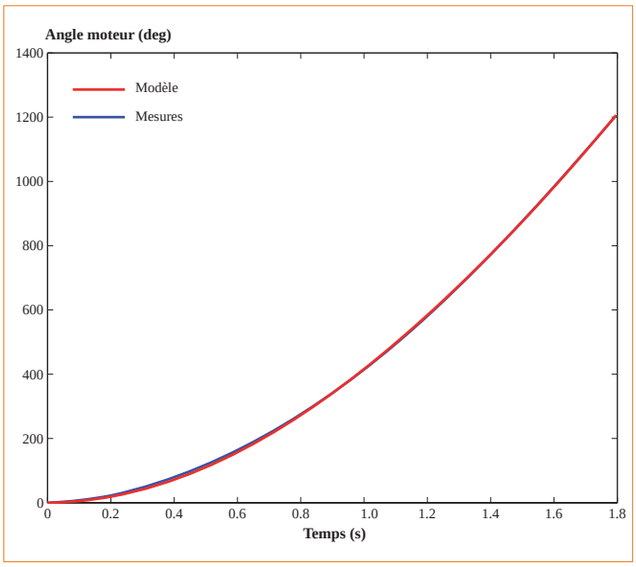
En intégrant, et en considérant nul l'angle moteur initial, on en déduit l'angle moteur en fonction du temps :

$$\theta_m(t) = \frac{1}{\mu}(MgR_p - C_s)\left(t + \frac{J_m}{\mu}\left(e^{-\frac{\mu}{J_m}t} - 1\right)\right)$$

Cette réponse du modèle proposé peut maintenant être comparée à la réponse expérimentale, et, éventuellement, une optimisation sur les paramètres mal connus du modèle peut être lancée de façon manuelle ou automatique à l'aide d'un logiciel (Matlab ou Excel). On obtient alors un jeu de paramètres optimal qui donne un comportement modèle relativement proche du comportement expérimental b :

$$J_m = 9{,}783 \times 10^{-3}\ \text{kg}\cdot\text{m}^2 \qquad \mu = 4{,}504 \times 10^{-3}\ \text{N}\cdot\text{m}\cdot\text{s}^{-1}$$

$$C_s = 1{,}588 \times 10^{-3}\ \text{N}\cdot\text{m}$$



b Les résultats du modèle dynamique de l'essai de lâcher après recalage et comparaison avec les résultats expérimentaux

La programmation pour tester

On souhaite réaliser un programme permettant de définir une variable interne K variant de 0 à K_{max} lorsque l'on tourne le potentiomètre d'une position extrême à l'autre.

Un premier programme permet d'afficher K à l'écran en le calculant à partir de la sortie brute du convertisseur S :

$$K = (K_{max} / S_{max})\ S$$

Une première série de mesures simples montre bien que le comportement n'est pas linéaire 17, ce qui n'est pas grave, mais pas pratique.

On peut également intégrer à ce programme la dépendance non linéaire de U_R à n 18, c'est-à-dire, une fois l'application numérique faite :

$$K = K_{max}\ (R_i / R_{max})\ [S / (1023 - S)]$$
$$= 10\ [S / (1023 - S)]$$

On obtient alors la seconde courbe de la figure 17, d'aspect bien plus linéaire que la précédente. Cela montre que les hypothèses faites sur le comportement sont correctes.

L'intégration dans le programme existant

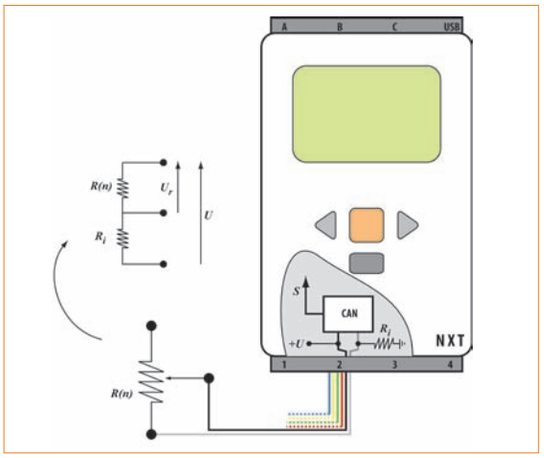
Pour intégrer le réglage des paramètres K_p et K_i dans le programme déjà réalisé, on a deux options :

- Le réglage est possible au démarrage du programme, et avant de lancer l'asservissement en position ; les variables K_p et K_i sont définies à ce moment-là, et il ne sera plus possible de les modifier par la suite.
- Le réglage est possible tout au long de l'exécution du programme ; les variables K_p et K_i sont mises à jour en permanence.

Conclusion et perspectives

Les travaux présentés ici ouvrent une voie permettant d'aborder de façon ludique et originale le monde des asservissements, mais aussi ceux de la dynamique et des transmissions mécaniques.

Bien d'autres projets, bien plus ambitieux, peuvent être consultés sur la Toile, et montrent combien l'engouement pour ces jouets supports est grand de la part des enseignants, du collège au supérieur, et ce, dans le monde entier. ■

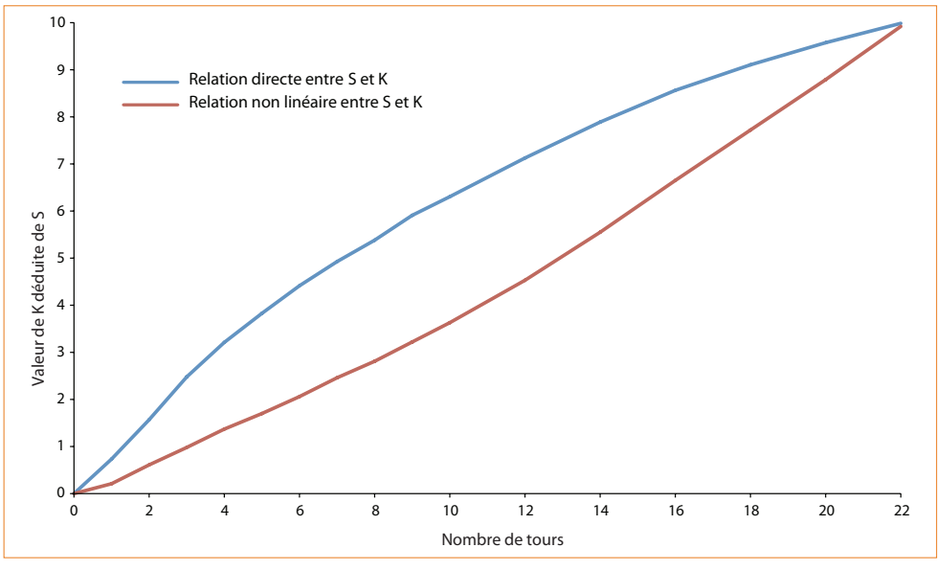


16 **Le détail du branchement interne du potentiomètre sur la brique NXT et schéma du pont diviseur équivalent**

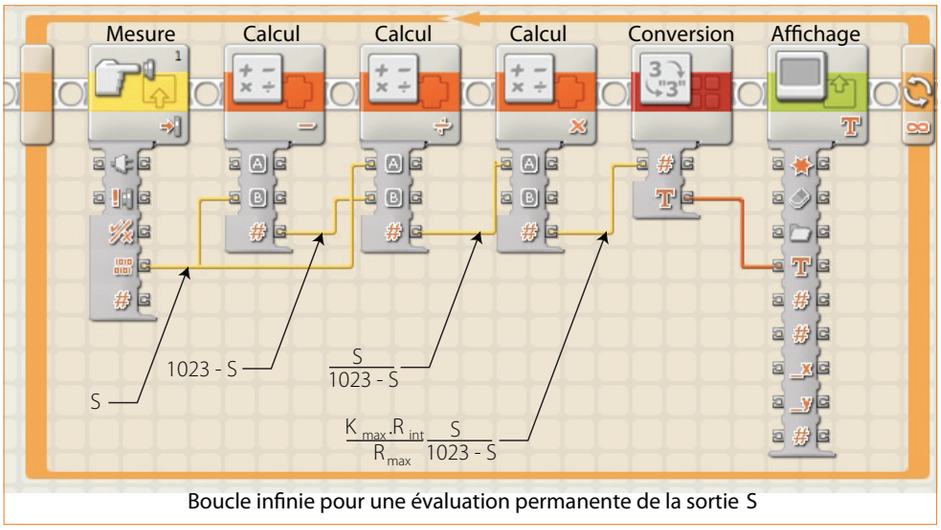
Références

[1] www.philohome.com/nxtmotor/nxtmotor.htm

[2] www.brickengineer.com



17 **L'évolution de la valeur du gain K en fonction du nombre de tours du potentiomètre**



Boucle infinie pour une évaluation permanente de la sortie S

18 **Le programme de calcul et d'affichage d'un gain K compris entre 0 et K_{max} variant linéairement avec le nombre de tours *n***