

# Le PID utilisé en régulation de position et/ou de vitesse de moteurs électriques<sup>1</sup>

CHRISTOPHE LE LANN<sup>2</sup>

2006-2007

<sup>1</sup>Contact extérieur : GAËL WAICHE, ingénieur, roboticien

<sup>2</sup>Pour plus d'informations : [www.totofweb.net](http://www.totofweb.net)

# Table des matières

<b>1 La théorie du PID appliquée aux moteurs électriques</b>	<b>3</b>
1.1 But à atteindre . . . . .	3
1.2 Modélisation mathématique d'un moteur électrique . . . . .	3
1.3 Boucle de retour . . . . .	5
1.4 Asservissements et influence des coefficients . . . . .	5
1.4.1 Asservissement P . . . . .	5
1.4.2 Asservissement PI . . . . .	6
1.4.3 Asservissement PID . . . . .	7
1.5 Réglage des coefficients . . . . .	8
1.6 Limitations et approximations . . . . .	9
<b>2 Application : Double asservissement d'un moteur</b>	<b>11</b>
2.1 Objectif de l'asservissement . . . . .	11
2.2 Modélisation du double asservissement . . . . .	11
2.3 Mise en oeuvre . . . . .	12
2.3.1 Partie mécanique . . . . .	12
2.3.2 Partie électronique . . . . .	12
2.3.3 Partie informatique embarquée . . . . .	13
2.3.4 Partie informatique distante . . . . .	15
2.4 Mise au point de l'asservissement - Résultats . . . . .	15
<b>3 Application : Asservissement en position et en vitesse d'un robot</b>	<b>18</b>
3.1 Objectif de l'asservissement . . . . .	18
3.2 Modélisation d'un robot à deux moteurs . . . . .	18
3.3 Mise en oeuvre . . . . .	19
3.3.1 Partie mécanique . . . . .	19
3.3.2 Partie électronique . . . . .	20
3.3.3 Partie informatique embarquée . . . . .	21
3.3.4 Partie informatique distante . . . . .	21
3.4 Mise au point de l'asservissement - Résultats . . . . .	21
<b>4 Conclusion</b>	<b>22</b>
4.1 Synthèse . . . . .	22
4.2 Alternatives à l'asservissement PID . . . . .	22

# Introduction

Dans la plupart des processus industriels, il est indispensable de maîtriser certains paramètres physiques. En automatique lorsque l'on souhaite atteindre une certaine vitesse, température, position, angle..., il est donc très souvent nécessaire d'avoir recours à un asservissement, c'est à dire un système capable d'atteindre et de maintenir une consigne en utilisant une mesure. Il s'agit donc d'un système bouclé, dont il reste à déterminer la fonction permettant de corriger la commande en fonction de la consigne initiale et de l'erreur mesurée.

Le PID est une méthode qui a fait ses preuves et qui donne de bons résultats. On peut citer, parmi tant d'autres, quelques exemples concrets d'utilisation : asservissement d'angle pour des bras robotiques (ABB, Kuka, Straübli, ...) et plus généralement dans les servomoteurs, régulation de réaction chimique en chimie pharmaceutique, convoyeurs industriels à navettes indexées, ...

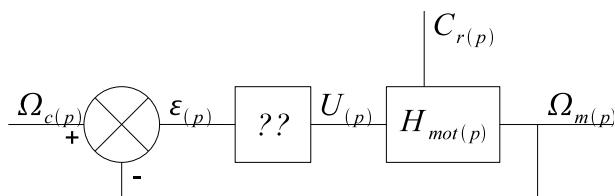


FIG. 1 –

(a) Schéma-bloc d'un moteur asservi en vitesse avec boucle de retour unitaire

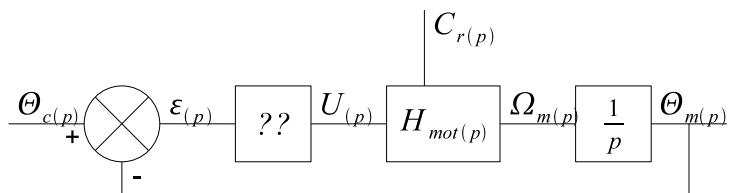


FIG. 2 –

(a) Schéma-bloc d'un moteur asservi en position avec boucle de retour unitaire



FIG. 3 –

(a) Un asservissement angulaire efficace donne aux bras robotiques KUKA une répétabilité de l'ordre du dixième de mm !

# Chapitre 1

# La théorie du PID appliquée aux moteurs électriques

L'un des critères d'efficacité d'un PID (et même de n'importe quel asservissement) est sa robustesse face aux changements : est-il encore capable de donner de bons résultats en cas de perturbations soudaines et/ou importantes ou de légères modifications du système (inertie, usure, ...) ?

## 1.1 But à atteindre

On cherche à faire atteindre une certaine valeur à une variable du système. Dans le cas d'un moteur, il s'agit principalement d'une vitesse ou d'une position.

- Les effets des perturbations doivent être minimisés, voire effacés, et ce le plus vite possible (régulation)
  - Les changements de consigne doivent être suivis rapidement et avec une bonne précision, si possible sans dépassement (poursuite)

Néanmoins, ces critères de performance sont souvent antagoniques, et nous verrons que le réglage de l'asservissement en détermine un compromis.

## 1.2 Modélisation mathématique d'un moteur électrique

Un moteur électrique à courant continu est régit par les équations physiques découlant de ses caractéristiques électriques, mécaniques et magnétiques. En utilisant le Théorème du Moment Cinétique et des équations d'électromagnétique, on obtient un système d'équations différentielles linéaires :

$$\left\{ \begin{array}{l} u(t) = e(t) + R.i(t) + L \frac{di(t)}{dt} \\ e(t) = K_e.\omega(t) \\ c_m(t) = K_c.i(t) \\ c_m(t) - c_r(t) = J_T \frac{d\omega(t)}{dt} \end{array} \right.$$

u Tension appliquée au moteur  
 e Force électromotrice  
 i Intensité traversant moteur  
 $\omega$  Vitesse de rotation du rotor  
 $c_m$  Couple moteur généré  
 $c_r$  Couple résistant

**Force électromotrice** : tension créée par le bobinage du moteur pour s'opposer à la variation du flux du champ magnétique le traversant.

On en déduit donc le modèle suivant dans le domaine de Laplace :

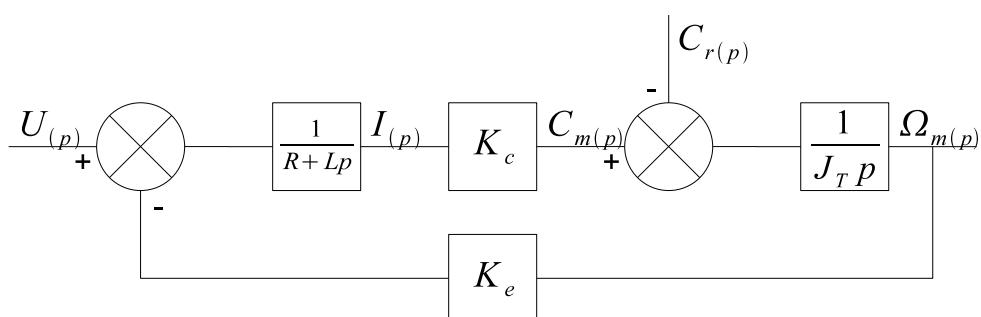


FIG. 1.1 -

(a) Schéma-bloc d'un modèle mathématique de moteur électrique

$$\Omega(p) = \frac{K_c}{K_e K_c + R J_T p + L J_T p^2} \cdot U(p) - \frac{R + L p}{K_e K_c + R J_T p + L J_T p^2} \cdot C_r(p)$$

La fonction de transfert associée à  $U$  possède les caractéristiques suivantes :

$\frac{\Omega(p)}{U(p)} = \frac{A}{1 + \frac{2\xi}{\omega_0} p + \frac{1}{\omega_0^2} p^2}$	$A = \frac{1}{K_e}$ $\xi = \frac{R}{2} \sqrt{\frac{J_T}{K_e K_c}}$ $\omega_0 = \sqrt{\frac{K_e K_c}{L J_T}}$	Gain statique Facteur d'amortissement Pulsation propre
---	--	--

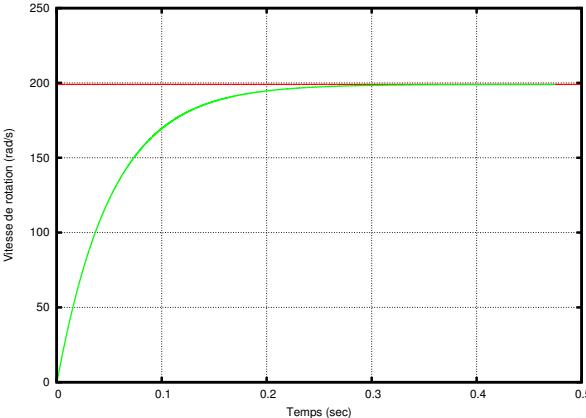


FIG. 1.2 –

(a) Modélisation sous Maple de la courbe de réponse à un échelon de tension

On observe que, bien que le système modélisé soit du second ordre, lorsque l'inductance interne est négligeable devant la résistance interne (ce qui est généralement le cas) il s'apparente à un système du premier ordre.

**Remarque** Les constantes  $K_e$  et  $K_c$ , si elles ne s'expriment pas dans la même unité ( $V.s$  ou  $N.m.A^{-1}$ ), représentent la même conversion de puissance électrique en puissance mécanique. Elles sont égales dans les unités du système international.

**Valeurs numériques** Les modélisations qui suivent sont basées sur les données constructeur d'un moteur Maxon :

Référence	F2260/813
Tension nominale	$U_{nom} = 24 V$
Tension maximale	$U_{max} = 32 V$
Courant permanent maximal	$I_{max} = 2.2 A$
Résistance aux bornes	$R = 1.91 \Omega$
Constante de couple	$K_c = 60.3 mNm.A^{-1}$
Constance de vitesse	$K_e = 60.3 mV.s$
Inductance	$L = 0.63 mH$
Moment d'inertie rapporté au rotor	$J_T = 10^{-4} kg.m^2$ (rotor seul)

**Frottements** Il est possible de modéliser sommairement les frottements internes au moteur par la somme deux types de frottements élémentaires :

- un frottement fluide : le couple résistant est proportionnel à la vitesse de rotation du rotor
- un frottement sec :
  - pour un couple moteur faible (inférieur au couple d'arrachement  $C_{arr}$ ), le couple résistant en est l'opposé, empêchant donc tout mouvement du rotor (le frottement fluide est alors nul)
  - pour un couple moteur dépassant le couple d'arrachement  $C_{arr}$ , le couple résistant est tronqué à une valeur seuil  $C_{sec}$  (le rotor peut alors tourner)

$$c_r(t) = \begin{cases} K_{fluide} \cdot \omega(t) + C_{sec} & \text{pour } c_m(t) > C_{arr} \\ c_m(t) & \text{pour } -C_{arr} \leq c_m(t) \leq C_{arr} \\ K_{fluide} \cdot \omega(t) - C_{sec} & \text{pour } c_m(t) < -C_{arr} \end{cases}$$

**Moment d'inertie rapporté au rotor** Dans le cas où le moteur met en mouvement un robot de masse  $M$ , il est possible de déterminer le nouveau moment d'inertie  $J_T$  perçu par le moteur sur son arbre en utilisant le Principe Fondamental de la Dynamique et le Théorème du Moment Cinétique et en connaissant le rayon  $R$  de la roue :

$$\left\{ \begin{array}{l} \sum \vec{F} = M \cdot \vec{a} \Rightarrow \frac{C_m}{R} = M \cdot a \\ \sum C = J_T \cdot \dot{\omega} \Rightarrow C_m = J_T \cdot \dot{\omega} \\ a = R \cdot \ddot{\omega} \end{array} \right. \Rightarrow J_T = M \cdot R^2$$

## 1.3 Boucle de retour

Étant donné que l'asservissement d'un moteur utilise une boucle fermée, il est nécessaire de disposer d'un capteur de position angulaire ou de vitesse angulaire. L'information renvoyée pourra être utilisée directement (cas d'une boucle à retour unitaire) ou traitée par calculs pour en déduire une information plus exploitable, comme la vitesse de déplacement d'un robot ou sa position, l'angle en sortie d'un réducteur, ...

Pour ce faire, plusieurs dispositifs existent, parmi lesquels :

- Encodeurs rotatifs incrémentaux (avec ou sans quadrature de phase, optique ou à effet hall) : capteurs équipés d'un dispositif optique ou magnétique émettant  $N$  impulsions par tour. Les plus élaborés permettent de déterminer le sens de rotation (deux signaux en quadrature, le déphasage donne le sens de rotation), voire même une position absolue.
- Tachogénératerices : capteurs renvoyant une tension analogique proportionnelle à la vitesse de rotation.
- Règles optiques : utilisent l'interférométrie optique pour obtenir une très haute résolution (supérieure à  $1\mu\text{m}$ ).
- Resolvers : utilisent des bobines en mouvement relatif produisant des signaux déphasés renseignant sur l'angle de l'arbre.

$$\varepsilon = \Omega_{\text{demande}} - \Omega_{\text{mesure}} \quad \text{ou} \quad \varepsilon = \Theta_{\text{demande}} - \Theta_{\text{mesure}}$$

## 1.4 Asservissements et influence des coefficients

### 1.4.1 Asservissement P

L'asservissement de type P est le plus simple qui soit. Il s'agit d'appliquer une correction proportionnelle à l'erreur corrigante de manière instantanée tout écart de la grandeur à régler :

$$\text{Consigne}(t) = K_p \cdot \varepsilon(t) \xrightarrow{\mathcal{L}} \text{Consigne}(p) = K_p \cdot \varepsilon(p)$$

Son rôle est d'amplifier virtuellement l'erreur pour que le système réagisse plus vivement, comme si l'erreur était plus grande qu'elle ne l'est en réalité.

Il permet de vaincre les grandes inerties du système et diminue le temps de montée en donnant de la puissance au moteur (plus l'erreur est grande, plus on donne de puissance au moteur). Lorsque l'on augmente  $K_p$ , le système réagit plus vite et l'erreur statique s'en trouve améliorée, mais en contrepartie le système perd en stabilité. Le dépassement se fait de plus en plus grand, et le système peut même diverger dans le cas d'un  $K_p$  démesuré.

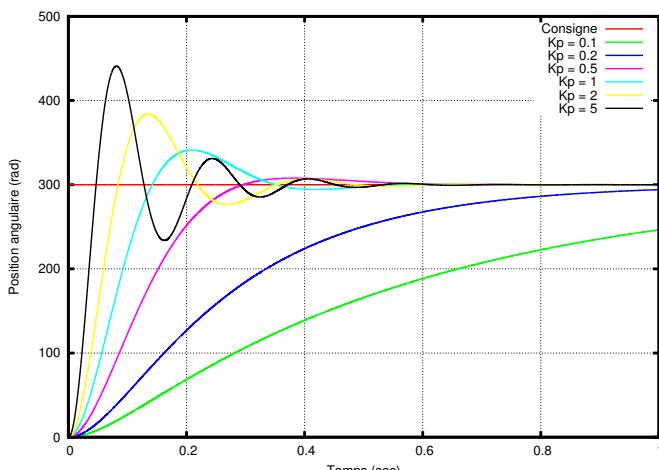


FIG. 1.3 –

(a) Modélisation sous Maple de la réponse à un échelon dans un asservissement en position

Néanmoins, il faut toujours une certaine tension aux bornes du moteur pour que celui-ci puisse tourner. C'est pourquoi, lorsque l'on s'approche de la valeur demandée, l'erreur n'est plus assez grande pour faire avancer le moteur, ce qui fait que l'on n'atteint jamais vraiment la valeur demandée. Il subsiste alors une erreur statique, qui est d'autant plus faible que  $K_p$  est grand.

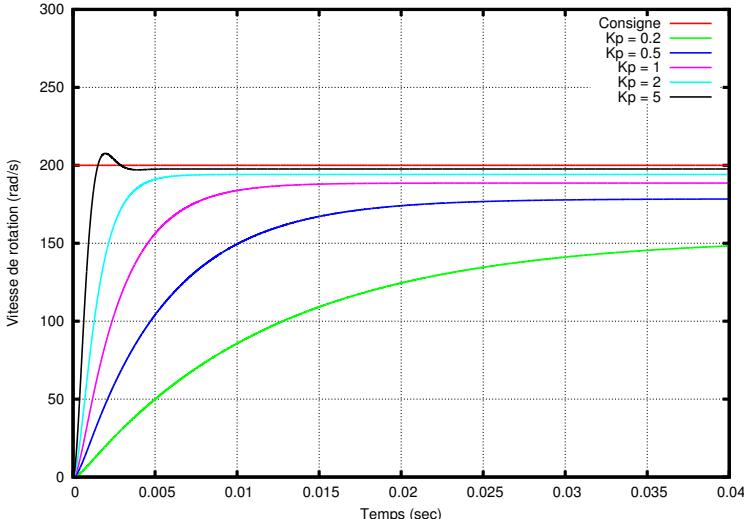


FIG. 1.4 –

(a) Modélisation sous Maple montrant la variation de l'erreur statique en fonction de  $K_p$  dans un asservissement en vitesse

### 1.4.2 Asservissement PI

L'asservissement de type PI est un asservissement de type P auquel on a ajouté un terme intégral :

$$Consigne(t) = K_p \cdot \varepsilon(t) + K_i \cdot \int_0^t \varepsilon(\tau) d\tau \quad \xrightarrow{\mathcal{L}} \quad Consigne(p) = K_p \cdot \varepsilon(p) + K_i \cdot \frac{\varepsilon(p)}{p} = \varepsilon(p) \cdot \left[ K_p + K_i \cdot \frac{1}{p} \right]$$

Le terme intégral complète l'action proportionnelle puisqu'il permet de compenser l'erreur statique et d'augmenter la précision en régime permanent. L'idée est d'intégrer l'erreur depuis le début et d'ajouter cette erreur à la consigne : lorsque l'on se rapproche de la valeur demandée, l'erreur devient de plus en plus faible. Le terme proportionnel n'agit plus mais le terme intégral subsiste et reste stable, ce qui maintient le moteur à la valeur demandée.

L'intégrale agissant comme un filtre sur le signal intégré, elle permet de diminuer l'impact des perturbations (bruit, parasites), et il en résulte alors un système plus stable. Malheureusement, un terme intégral trop important peut lui aussi entraîner un dépassement de la consigne, une stabilisation plus lente, voire même des oscillations divergentes.

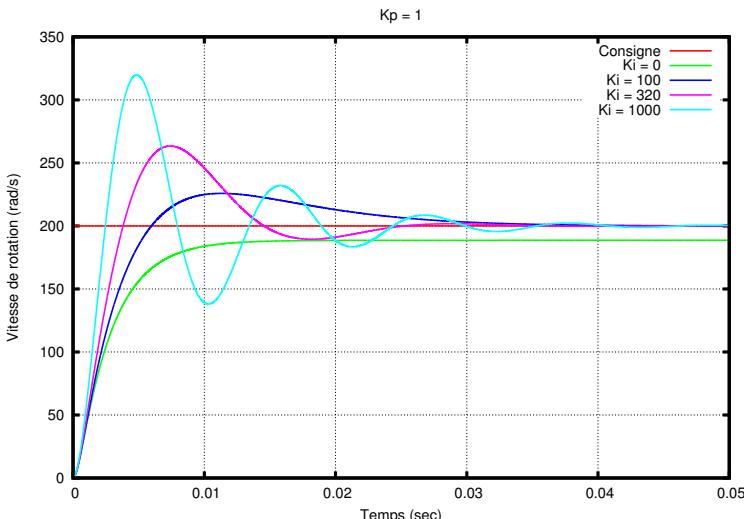


FIG. 1.5 –

(a) Modélisation sous Maple de la réponse à un échelon dans un asservissement en vitesse

### 1.4.3 Asservissement PID

Les termes proportionnel et intégral peuvent amener un dépassement de la consigne et des oscillations. Cela implique pour le moteur des inversions de polarité, ce qui est loin d'être idéal. Pour limiter ce phénomène indésirable, on introduit un troisième élément : le terme dérivé. Son action va dépendre du signe et de la vitesse de variation de l'erreur, et sera opposée à l'action proportionnelle. Elle devient prépondérante aux abords de la valeur demandée lorsque l'erreur devient faible, que l'action du terme proportionnel faiblit et que l'intégrale varie peu : elle freine alors le système, limitant le dépassement et diminuant le temps de stabilisation.

$$Consigne(t) = K_p \cdot \varepsilon(t) + K_i \cdot \int_0^t \varepsilon(\tau) d\tau + K_d \cdot \frac{d}{dt} \varepsilon(t)$$

$$Consigne(p) = K_p \cdot \varepsilon(p) + K_i \cdot \frac{\varepsilon(p)}{p} + K_d \cdot p \cdot \varepsilon(p) = \varepsilon(p) \cdot \left[ K_p + K_i \cdot \frac{1}{p} + K_d \cdot p \right]$$

On obtient alors un asservissement composée d'un terme proportionnel, un terme intégral et un terme dérivé : on parle d'asservissement PID.

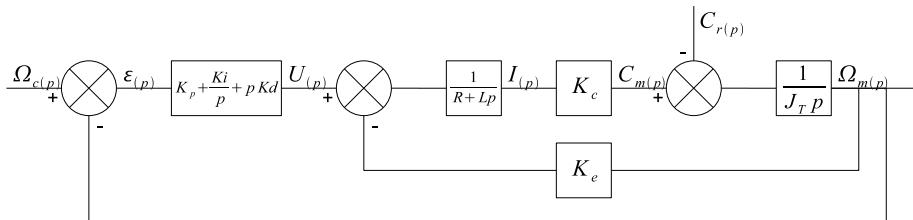


FIG. 1.6 –

(a) Schéma-bloc de l'implantation du PID en régulation de vitesse

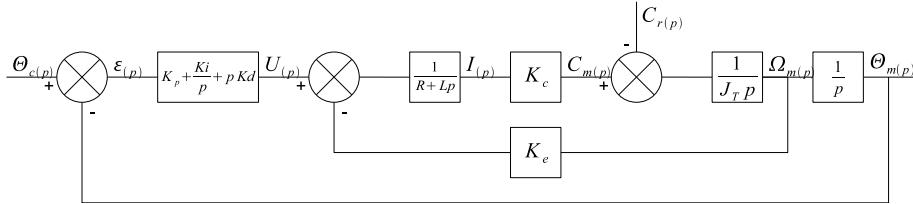


FIG. 1.7 –

(a) Schéma-bloc de l'implantation du PID en régulation de position

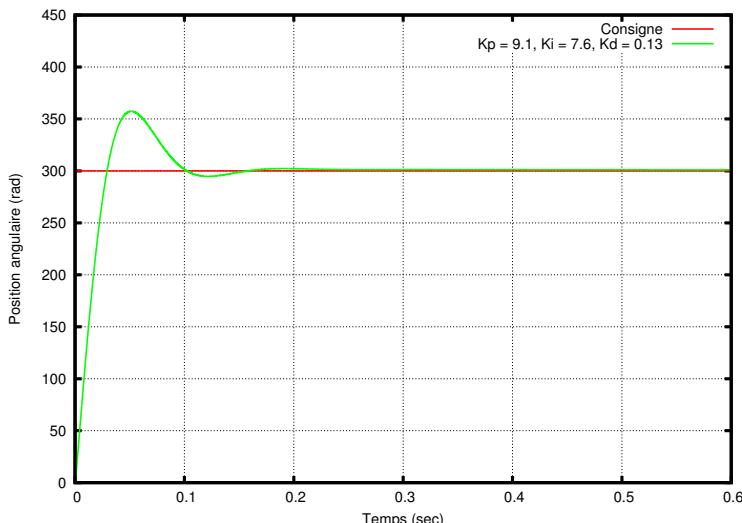


FIG. 1.8 –

(a) Modélisation sous Maple de la réponse à un échelon dans un asservissement en position

**Remarque** L'action dérivée est surtout utilisée dans le cas de variables non bruitées, car la dérivation est très sensible au bruitage du signal : on diminuera donc son influence dans un asservissement de vitesse, pour lequel la dérivée est l'accélération, variable soumise à de nombreuses perturbations.

## Récapitulatif de l'action des coefficients

Coefficient	Temps de montée	Temps de stabilisation	Dépassemement	Erreur statique
$K_p$	Diminue	Augmente	Augmente	Diminue
$K_i$	Diminue	Augmente	Augmente	Annule
$K_d$	-	Diminue	Diminue	-

## 1.5 Réglage des coefficients

Le réglage d'un PID consiste à trouver les meilleurs coefficients  $K_p$ ,  $K_i$  et  $K_d$  dans le but d'obtenir une réponse adéquate du procédé et de la régulation. L'objectif est d'être robuste, rapide et précis tout en limitant les dépassemements.

### Méthode de Ziegler-Nichols

La méthode empirique de Ziegler-Nichols est très répandue dans l'industrie ou chez les automatiens pour régler les correcteurs de ce type de chaîne. Elle a l'avantage de ne pas nécessiter de modélisation précise du système asservi, mais se contente d'essais expérimentaux, ce qui rend cette méthode très simple.

On fixe  $K_i$  et  $K_d$  à 0 et on augmente le gain du correcteur proportionnel pur  $K_p$  jusqu'à obtenir des oscillations persistantes (pour des moteurs, il s'agira plutôt d'une vibration autour de la position demandée). On mesure alors  $(K_p)_{lim}$ , le gain du correcteur proportionnel à la juste oscillation, et  $T_{osc}$ , la période des oscillations.

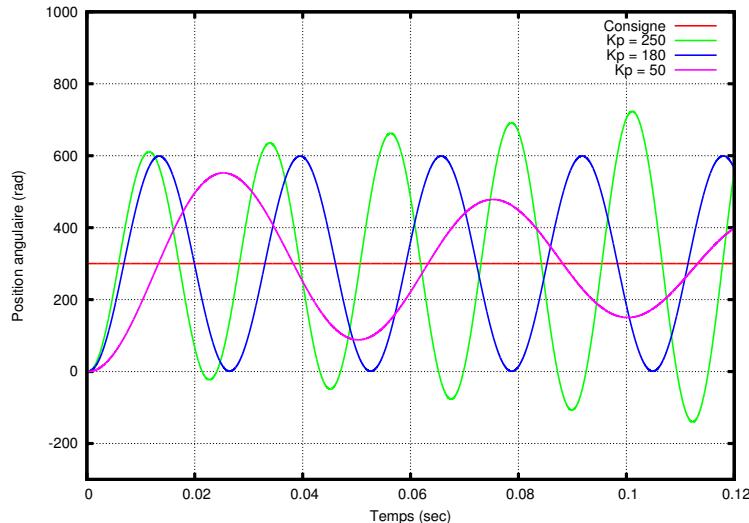


FIG. 1.9 –

(a) Modélisation sous Maple montrant l'apparition d'oscillations dans un asservissement en position de type P

La méthode de Ziegler-Nichols indique alors des valeurs des coefficients permettant d'obtenir un bon compromis entre précision, rapidité et stabilité.

$$\begin{cases} K_p = 0.6 \cdot (K_p)_{lim} \\ K_i = \frac{1}{0.5 \cdot T_{osc}} \\ K_d = 0.125 \cdot T_{osc} \end{cases}$$

Ensuite, on peut légèrement faire varier ces valeurs moyennes selon le compromis précision/rapidité/stabilité recherché. Dans le cas d'un asservissement en position, on aura tendance à monter un peu plus le gain intégral pour avoir une erreur statique faible, quitte à « accoster » plus lentement à la valeur demandée.

**Remarque** Il ne s'agit ici que de la seconde méthode de Ziegler-Nichols, utilisant des mesures expérimentales en boucle fermée et une conception continue du temps (corrections en temps réel). Il existe aussi d'autres méthodes de Ziegler-Nichols : l'une est aussi basée sur des mesures en boucle fermée, mais dans une conception discrète du temps (N corrections par seconde, il faut alors prendre en compte le retard de correction), l'autre est basée sur des mesures expérimentales en boucle ouverte.

Il se peut aussi que le système soit désespérément stable. La méthode de Ziegler-Nichols n'est alors pas nécessaire : puisque l'on peut se permettre d'augmenter  $K_p$  à volonté, on choisit une valeur relativement grande de  $K_p$ , puis on ajuste  $K_i$  et  $K_d$  pour éliminer l'erreur statique, limiter le dépassemement et atteindre la consigne le plus vite possible.

Néanmoins, il faut aussi tenir compte du fait qu'un moteur n'accepte pas de trop fortes tensions : en pratique on ne peut augmenter les coefficients à volonté sans arriver à ce seuil de commande : le signal corrigé pourra éventuellement se retrouver tronqué, et le modèle considéré jusqu'ici ne sera alors plus valable.

## Méthode avancée

Une méthode plus avancée consiste à modéliser parfaitement le système asservi, mais nécessite une connaissance parfaite de l'ensemble des paramètres permettant d'obtenir une modélisation précise. On s'applique alors à chercher à optimiser le temps de réponse et la stabilité par une étude mathématique.

L'avantage est que cette méthode permet en théorie d'obtenir les réglages les plus fin possible. Mais l'inconvénient est qu'il est parfois difficile de modéliser parfaitement un système lorsque de nombreux phénomènes parasites (jeu mécanique dans les réducteurs, frottements et pertes omniprésents, usure, ...) ne sont pas négligeables. La précision des paramètres déterminés est alors à l'image de celle de la modélisation.

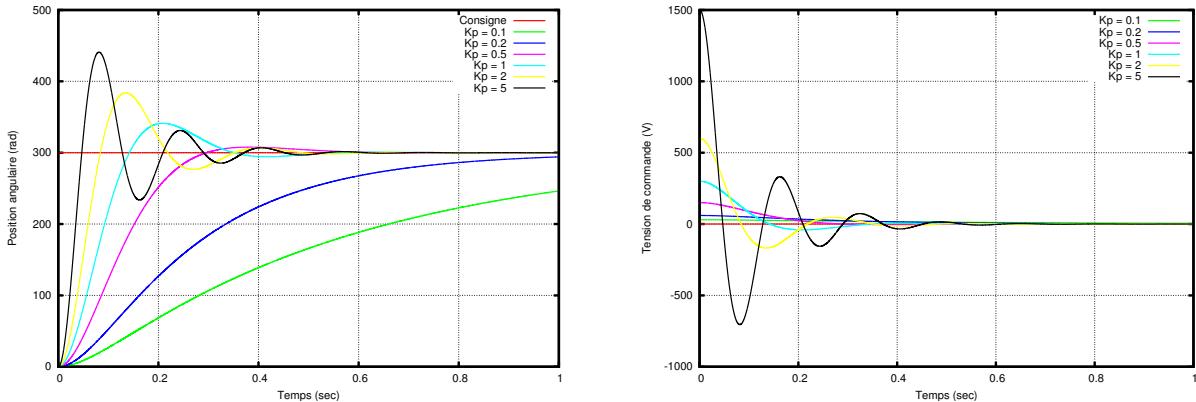
**Remarque** On peut aussi tester directement la réponse du système à des signaux particuliers et en déduire une forme approchée de la fonction de transfert, puis se baser sur ce modèle approché pour mener à bien les calculs mathématiques.

## 1.6 Limitations et approximations

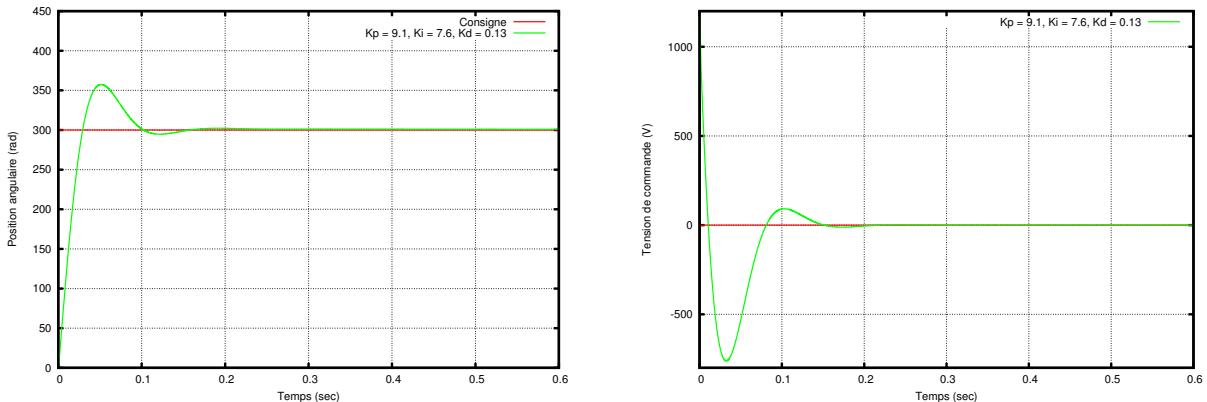
### Limitation de la tension de commande

Lors du réglage des coefficients du PID, il est primordial de se rappeler qu'un moteur n'est utilisable que dans une certaine plage de tensions ( $\pm 32V$  dans notre cas). Hors de cette zone, non seulement le comportement du moteur ne suivra plus obligatoirement le modèle établi, mais le moteur pourra éventuellement se détériorer irrémédiablement.

Nous ne sommes donc pas totalement libres de choisir les coefficients  $K_p$ ,  $K_i$  et  $K_d$ . Voici des modélisations sous Maple illustrant ce problème sur un asservissement en position de type P :



Il est donc probable que le réglage optimal théorique ne puisse être atteint, ce qui fut notre cas :



D'autre part, les frottements secs n'ont pas pu être pris en compte lors de cette modélisation étant donné qu'ils n'étaient pas transposables dans le domaine de Laplace. Il est probable que leur modélisation aurait entraîné deux principales conséquences qui n'ont pas pu être décelées lors de cette modélisation sous Maple :

- Un moteur n'est pas sensé démarrer pour des tensions trop faibles en raison de ce frottement sec.
- L'erreur statique aurait probablement été plus grande.

## Approximations

Le choix du modèle de moteur peut induire certaines approximations :

- Le jeu mécanique des pièces (notamment des réducteurs s'il y en a) n'est pas pris en compte.
- L'usure des pièces n'est pas prise en compte.
- Le modèle n'est valable que pour une plage de tension et de courant au-delà de laquelle le comportement et la durée de vie du moteur sont incertains.

Le choix de la méthode d'implantation du PID peut elle aussi induire certaines approximations :

- Approximation des calculs du nombre dérivé et de l'intégrale (calculs approchés dans une implantation logicielle, non-linéarité dans une implantation par AOP, ...).
- Approximation en considérant le temps comme discret et non continu (fréquence de recalculation de la correction dans une implantation logicielle).
- Approximation au niveau des capteurs (nombre de pas des encodeurs, mauvais comptage de pas, conversion analogique-numérique des tachogénératerices, ...).

Finalement, l'erreur est faible, mais il est nécessaire d'en tenir compte. Pour minimiser son effet, diverses solutions sont envisageables :

- Utiliser un système hardware de dérivation et d'intégration (lequel reste soumis aux approximations des CAN).
- Diminuer  $\Delta t$  pour se rapprocher d'un modèle (idéal) où le temps est continu.
- Améliorer la précision des capteurs.
- Recalculer régulièrement le système (signal d'index sur les encodeurs, réaffectation des coordonnées en un point de coordonnées connues).

# Chapitre 2

## Application : Double asservissement d'un moteur

### 2.1 Objectif de l'asservissement

L'objectif d'un double asservissement, à la fois en position et en vitesse, est de maîtriser la vitesse de déplacement dans un asservissement de position. Cela est notamment mis en oeuvre dans les bras robotiques (Kuka, Straübli, ABB, ...) et dans de nombreuses machines à commande numérique.

L'avantage est que la maîtrise de la vitesse de déplacement permet de définir une courbe de vitesse au cours d'un déplacement indépendamment de la distance à parcourir. En effet, un simple asservissement PID de position modifie la tension à appliquer selon la distance du déplacement. Or il est souvent préférable de se déplacer à une vitesse globalement constante quelle que soit la distance à parcourir, en accélérant progressivement jusqu'à un palier, puis en décélérant juste avant d'atteindre la position finale afin d'éviter de la dépasser (ce que fait naturellement un conducteur de véhicule). Typiquement, on implémente une commande trapézoïdale, c'est-à-dire que l'on applique un seuil à la fois sur la vitesse et l'accélération :

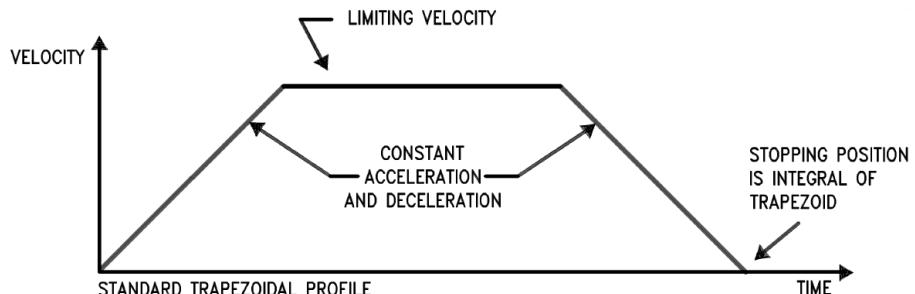
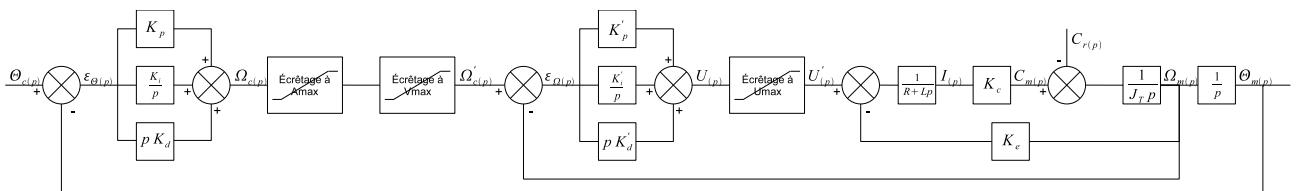


FIG. 2.1 –

(a) Profil de vitesse trapézoïdal mis en place par un LM629 (composant électronique spécialisé)

Un autre intérêt est que le contrôle de l'accélération évite les mouvements trop brusques, notamment lors de la manipulation d'objets délicats. Le contrôle de la vitesse uniformise tous les déplacements et évite d'atteindre des vitesses trop élevées (le moteur est alors contrôlé dans sa plage de tensions nominale).

### 2.2 Modélisation du double asservissement



(b) Schéma-bloc de la double implémentation de PID tenant compte des écrétages

On peut donc décomposer ce double asservissement en deux fonctions de régulation, une première régulant la position et une seconde régulant la vitesse sur les ordres de la première :

- La première est un asservissement de position de type PID. Sa particularité est d'être sujet à posteriori à deux contraintes (malheureusement non transposables dans le domaine de Laplace), l'une concernant l'accélération maximale et l'autre concernant la vitesse maximale.
- La seconde est un asservissement de vitesse de type PID comme étudié au chapitre précédent.

Néanmoins le choix des coefficients du PID de position n'est pas totalement libre : étant soumis au seuillage de l'accélération, si le PID est réglé pour monter très vite en vitesse et décélérer tard avant d'accoster à la position finale il subira l'action du seuillage d'accélération. Le moteur ne ralentira pas assez vite et dépassera sa position finale, engendrant petit à petit des oscillations divergentes. Il faut donc choisir les coefficients suffisamment faibles pour ralentir assez tôt, ce qui est d'autant plus contraignant que l'accélération maximale choisie est faible.

## 2.3 Mise en oeuvre

### 2.3.1 Partie mécanique

Un encodeur optique incrémental à quadrature de phase est relié au rotor du moteur, ce qui nous permet de récupérer une information sur l'incrémentation angulaire du rotor, à partir de laquelle on peut calculer la vitesse de rotation du rotor et sa position angulaire par rapport à sa position initiale. Il s'agit d'un disque en rotation doté de deux rangées de fentes décalées. Un système optique détecte ces fentes et émet 512 impulsions par tour de rotor, sur deux signaux en quadrature de phase (la quadrature servant ici à déterminer le sens de rotation du rotor).

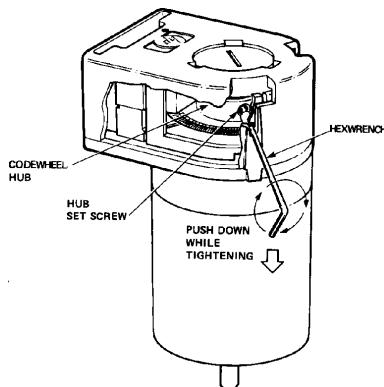


FIG. 2.2 –  
(a) Utilisation d'un encodeur optique  
sur l'axe du moteur

### 2.3.2 Partie électronique

Deux cartes électroniques sont utilisées pour ce double asservissement :

- Une première carte s'occupe de l'acquisition et du traitement des signaux. Elle s'articule autour d'un micro-contrôleur<sup>1</sup> (AtMega16 à 16MHz) qui permet une grande souplesse dans la programmation en C de l'asservissement. Ce micro-contrôleur acquiert les signaux venant de l'encodeur optique par l'intermédiaire d'une bascule D et d'un diviseur de fréquence (des circuits intégrés qui se chargent d'un prétraitement des signaux en quadrature), et calcule les corrections à appliquer en fonction de la commande initiale. Une liaison série permet de tester diverses valeurs pour les coefficients du PID et d'en récupérer les courbes de réponse (vitesse et/ou position) sur un ordinateur.
- Une carte puissance, basée sur un double pont en H<sup>2</sup>, permet de contrôler la tension à fournir aux moteurs (par une modulation PWM) ainsi que le sens de déplacement selon les ordres de la carte principale. Une modulation PWM (Pulse-Width Modulation, ou Modulation de Largeur d'Impulsion) permet d'obtenir un contrôle proportionnel sur un moteur sans avoir à varier la tension de commande, selon le principe du hacheur électrique :

$$\begin{cases} V = V_{max} & \text{pour } 0 < t < \alpha T \\ V = 0 & \text{pour } \alpha T < t < T \end{cases} \quad \text{avec } 0 < \alpha < 1 \quad \Rightarrow \quad V_{moy} = \alpha \cdot V_{max}$$

<sup>1</sup> Composant électronique numérique intégrant entre autres un processeur, une mémoire programme, une mémoire RAM et des ports d'entrée/sortie, pouvant être programmé afin d'obtenir un comportement personnalisé.

<sup>2</sup> Architecture permettant de contrôler un moteur dans ses deux sens de fonctionnement tout en permettant une commutation rapide.

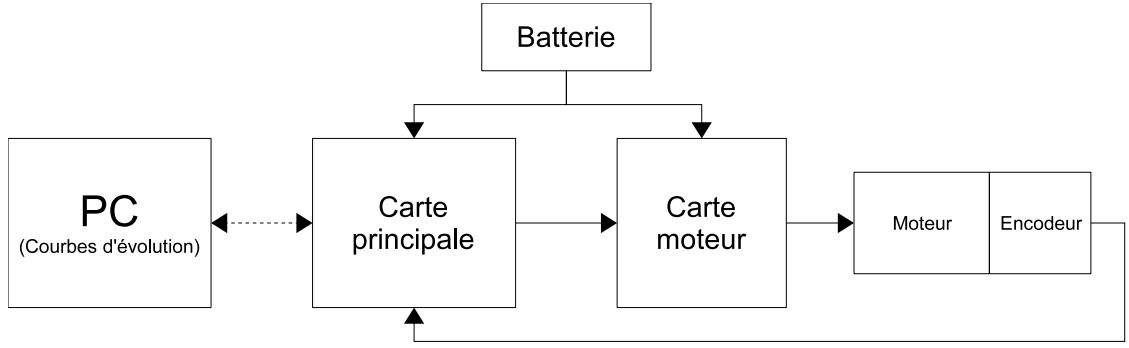


FIG. 2.3 –

### 2.3.3 Partie informatique embarquée

#### Acquisition des données

A intervalle de temps régulier (toutes les 50ms), le micro-contrôleur lit la position angulaire donnée par le capteur et met à jour ses variables internes de position et de vitesse. La mesure des erreurs (erreur de position et erreur de vitesse) se fait par une simple différence entre la valeur demandée et la valeur mesurée. C'est la fréquence de rafraîchissement des erreurs et de recalculation de la correction qui joue un rôle significatif dans la précision de l'asservissement :

- Si la fréquence de rafraîchissement est trop faible, le trop grand écart entre deux recalculs de correction opère une discréétisation du temps trop grossière, ce qui amoindrira la précision de l'asservissement.
- Si la fréquence de rafraîchissement est trop élevée, bien que l'on se rapproche du modèle de temps continu la précision de l'asservissement s'amoindrira puisque le nombre de pas comptés par le capteur entre deux recalculs de correction sera plus faible, donc moins précis.
- Une fréquence de 20Hz a été jugée empiriquement comme un assez bon compromis entre période d'échantillonnage et précision de chaque recalculation lors des expériences.

#### Asservissement de vitesse

L'essentiel de l'implantation du PID est réalisé informatiquement, par programmation en C du micro-contrôleur intégré à la carte principale. Il s'agit de traduire par un algorithme le calcul en continu de la tension à envoyer au moteur selon l'erreur mesurée à partir de l'encodeur optique.

**Calcul approximé de la dérivée** On estime le nombre dérivé en tenant compte des deux dernières mesures par une approximation au premier ordre de la fonction erreur  $\varepsilon$  à l'instant considéré. Étant donné que  $\Delta t$  (temps entre deux mesures,  $\Delta t = 50\text{ms}$  à  $20\text{Hz}$ ) est assez petit par rapport à la vitesse de variation du système, on considère qu'il s'agit là d'une approximation convenable.

$$\frac{d}{dt}\varepsilon(t) = \lim_{\Delta t \rightarrow 0} \frac{\varepsilon(t) - \varepsilon(t - \Delta t)}{\Delta t} \simeq \frac{\varepsilon(t) - \varepsilon(t - \Delta t)}{\Delta t}$$

**Remarque** Pour éviter de surcharger le micro-contrôleur avec une division en virgule flottante à  $20\text{Hz}$ , on inclut le terme  $\frac{1}{\Delta t}$  directement dans le terme  $K_d$  paramétré.

**Calcul approximé de l'intégrale** On utilise la formule de Riemann pour obtenir une approximation de l'intégration de l'erreur par la somme discrète des erreurs mesurées :

$$\int_a^b f(t) dt = \lim_{n \rightarrow \infty} \frac{b-a}{n} \sum_{k=0}^n f(a + k \cdot \frac{b-a}{n}) \Rightarrow \int_0^t \varepsilon(\tau) d\tau = \lim_{\Delta t \rightarrow 0} \Delta t \sum_{k=0}^{t/\Delta t} \varepsilon(k \cdot \Delta t) \simeq \Delta t \sum_{k=0}^{t/\Delta t} \varepsilon(k \cdot \Delta t)$$

**Remarque**  $a = 0$ ,  $b = t$ ,  $n = \frac{t}{\Delta t}$

On se contente donc de sommer l'ensemble des erreurs mesurées, ce qui donne le terme  $\sum_{k=0}^{t/\Delta t} \varepsilon(k \cdot \Delta t)$ . Le terme  $\Delta t$  sera déjà inclus logiciellement dans le paramètre  $K_i$  pour éviter de surcharger le micro-contrôleur avec une multiplication en virgule flottante supplémentaire à  $20\text{Hz}$ .

**Calcul du PID** A chaque nouvelle mesure de l'erreur, on recalcule la consigne corrigée à envoyer aux moteurs après avoir effectué les calculs préliminaires de dérivation et d'intégration :

$$\begin{aligned} Consigne(t) &= K_p \cdot \varepsilon(t) + K_i \cdot \int_0^t \varepsilon(\tau) d\tau + K_d \cdot \frac{d}{dt} \varepsilon(t) \\ Consigne(p) &= K_p \cdot \varepsilon(p) + K_i \cdot \frac{\varepsilon(p)}{p} + K_d \cdot p \cdot \varepsilon(p) = \varepsilon(p) \cdot \left[ K_p + K_i \cdot \frac{1}{p} + K_d \cdot p \right] \end{aligned}$$

**Code final simplifié** Il ne s'agit ici que des « grandes lignes » (simplifiées), montrant le principe de base sans pour autant suffire à mettre en oeuvre le PID.

```
// VKp, VKi et VKd sont définis statiquement par des « #define »
// PID et Encodeur sont des structures globales
void PID_Recalc(void) {
    // Acquisition des données et mesure de l'erreur
    PID.vitesse = Encodeur.position - PID.position_old; // Exprimé en « pas par DeltaT »
    PID.position_old = Encodeur.position;
    // Mesure de l'erreur
    PID.erreur_old = PID.erreur;
    PID.erreur = PID.consigne_vitesse - PID.vitesse;
    // Calcul du nombre dérivé (sauf le terme 1/DeltaT intégré dans VKd)
    PID.derive = PID.erreur - PID.erreur_old;
    // Calcul du terme intégral (sauf le terme DeltaT intégré dans VKi)
    PID.integral += PID.erreur;
    // Calcul de la nouvelle commande
    PID.pid = VKp*PID.erreur + VKi*PID.integral + VKd*PID.derive;
    // Actualisation de la PWM envoyée au moteur
    Moteur_PWM(PID.pid);
}
```

**Remarques** Cette méthode d'implantation du PID implique d'approximer sciemment les calculs de dérivée et d'intégrale, ainsi que les résultats globaux (la sortie de commande du moteur n'est échelonnée « que » sur 10 bits). Néanmoins, les résultats montrent que ces approximations n'ont pas trop dégradé la précision du système.

## Asservissement de position et seuillage de l'accélération et de la vitesse

L'asservissement de position est géré d'une manière tout à fait similaire à l'asservissement de vitesse. La principale différence est que sa variable de sortie sert de consigne à l'asservissement de vitesse après avoir été traitée algorithmiquement pour prendre en compte la vitesse maximale et de l'accélération maximale.

```
// Appelé par interruption3 tous les DeltaT
// PID et Encodeur sont des structures globales, PKp, PKi, PKd, Amax et Vmax sont définis statiquement
// T est une variable temporelle
void ProfilVitesse() {
    PID.consigne_vitesse_old = PID.consigne_vitesse; // Sauvegarde de l'ancienne consigne
    PID.erreur_position_old = PID.erreur_position; // Sauvegarde de l'ancienne erreur
    PID.erreur_position = PID.consigne_position - Encodeur.position; // Calcul de la nouvelle erreur
    PID.integral_position += PID.erreur_position; // Recalcul de l'intégrale
    PID.consigne_vitesse = PKp*PID.erreur_position + PKi*PID.integral_position + PKd*(PID.erreur_position - PID.erreur_position_old);
    // Écrêtage de la vitesse
    if (PID.consigne_vitesse > Vmax) {
        PID.consigne_vitesse = Vmax;
    } else if (PID.consigne_vitesse < -Vmax) {
        PID.consigne_vitesse = -Vmax;
    }
    // Écrêtage de l'accélération
    if ((PID.consigne_vitesse - PID.consigne_vitesse_old) > Amax) {
        PID.consigne_vitesse = PID.consigne_vitesse_old + Amax;
    } else if ((PID.consigne_vitesse - PID.consigne_vitesse_old) < -Amax) {
        PID.consigne_vitesse = PID.consigne_vitesse_old - Amax;
    }
}
```

<sup>3</sup>Ici, une interruption est un évènement logiciel survenant à intervalle de temps fixe, qui interrompt le cours normal d'un programme, exécute une courte tâche prioritaire, puis reprends le cours du programme principal.

```

    }
    PID_ReCalc() ;
}

```

### 2.3.4 Partie informatique distante

Pour pouvoir régler l'asservissement, il a été nécessaire de récupérer l'historique d'évolution de diverses variables et d'en tracer la représentation graphique après adaptation des données brutes en un format plus exploitable.

1. Au début de chaque session de mesures, le micro-contrôleur a demandé à l'ordinateur distant (via une connexion série) les paramètres à utiliser : coefficients du/des PID, consigne à atteindre, vitesse maximale, accélération maximale et durée de session. Une fois ces paramètres renseignés, la session de mesures pouvait commencer.
2. Dès lors, en plus de gérer le double asservissement du moteur, le micro-contrôleur a envoyé à l'ordinateur (via la même connexion série) les variables intéressantes, et ce à chaque boucle de correction : position, vitesse, erreur, correction demandée et dépassement de tension (lorsque la tension exigée par le PID dépassait la tension admissible par le moteur).
3. Une fois ces données enregistrées, elles ont été confiées à un logiciel spécialisé dans l'exploitation de données et le tracé de courbes (Gnuplot) afin d'en tracer la représentation graphique de manière facilitée (par une ligne de commande sous Linux) après avoir traité les données brutes pour les convertir en unités intéressantes (les données reçues étaient exprimées en « pas » ou en « pas par DeltaT »).
4. Une fois la session terminée, les variables internes (état de l'intégrale, position angulaire, ...) ont été réinitialisées avant de redemander les paramètres à utiliser pour une nouvelle session.

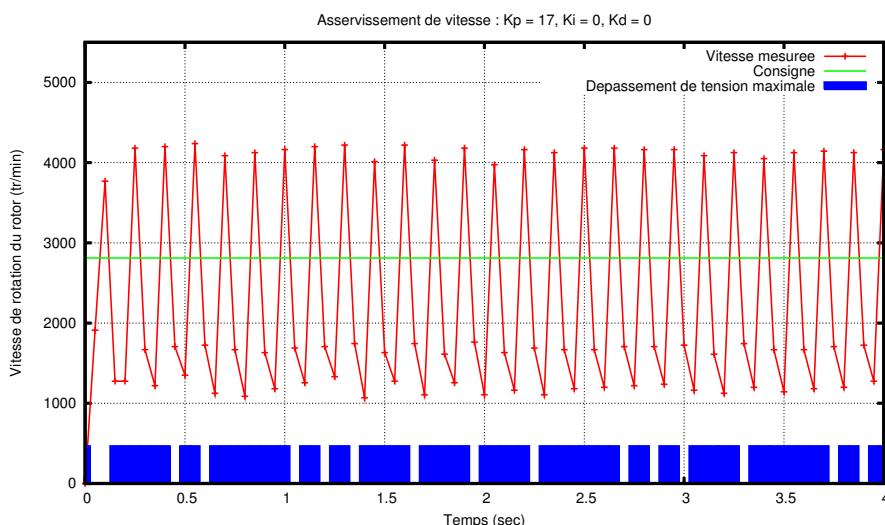
Lorsqu'il a été nécessaire d'opérer à une batterie complète de mesures, un script a été utilisé du côté de l'ordinateur afin de renseigner automatiquement les paramètres de session à la carte principale et d'exploiter les données reçues de manière autonome.

## 2.4 Mise au point de l'asservissement - Résultats

### Asservissement de vitesse

Étant donné que les moteurs à ma disposition étaient d'une qualité bien moindre que celle des moteurs hauts de gamme (Escap, Maxon, ...) généralement utilisés pour les asservissements nécessitant de la précision, la plupart des paramètres (inductance, constante de couple, ...) sont inconnus et difficilement déterminables. Plutôt que d'essayer de les déterminer via des expériences complexes, la méthode de Ziegler-Nichols a été jugée plus adaptée à la mise au point de l'asservissement.

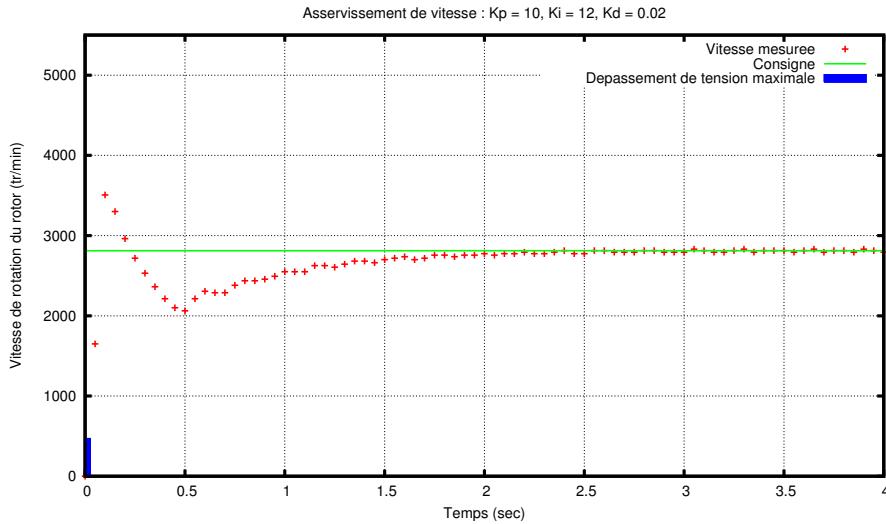
Cette méthode a donc nécessité d'augmenter le gain proportionnel  $K_p$  du PID (avec  $K_i$  et  $K_d$  fixés à 0) jusqu'à obtenir une oscillation sur la vitesse. Malheureusement, étant donné que la période des oscillations a été faible devant la période d'échantillonage, il a été difficile d'observer de franches oscillations sinusoïdales, mais plutôt des vibrations régulières autour du point d'équilibre. Voici la courbe de réponse mesurée par le micro-contrôleur pour la valeur de  $K_p$  provoquant des oscillations maintenues :



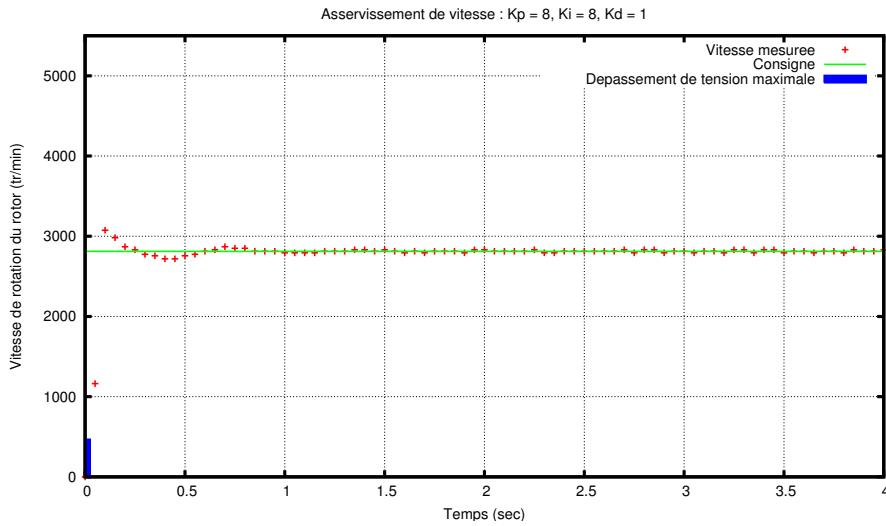
D'après la méthode de Ziegler-Nichols, on en déduit donc :

$$\left\{ \begin{array}{l} (K_p)_{lim} = 17 \\ T_{osc} = 0.15s \end{array} \right. \Rightarrow \left\{ \begin{array}{l} K_p = 10.2 \\ K_i = 13.3 \\ K_d = 18.7 \cdot 10^{-3} \end{array} \right.$$

Néanmoins, comme on pouvait s'y attendre la valeur de  $K_p$  provoquant les oscillations maintenues est trop grande, ce qui nécessiterait donc de suralimenter le moteur au-delà de sa plage de tension nominale. La tension prévue par le PID est donc tronquée à la valeur maximale acceptée par le moteur (ce qui est signalé sur le graphique par les marqueurs bleus), le modèle théorique n'est plus suivi parfaitement : il y a une marge d'erreur dans la détermination de  $(K_p)_{lim}$  et de  $T_{osc}$ , et il va donc falloir modifier les valeurs optimales théoriques pour affiner le réglage. Voici la courbe de réponse mesurée pour le réglage proposé par la méthode de Ziegler-Nichols :



Une batterie de mesures automatisées a donc permis d'obtenir un réglage plus optimal :



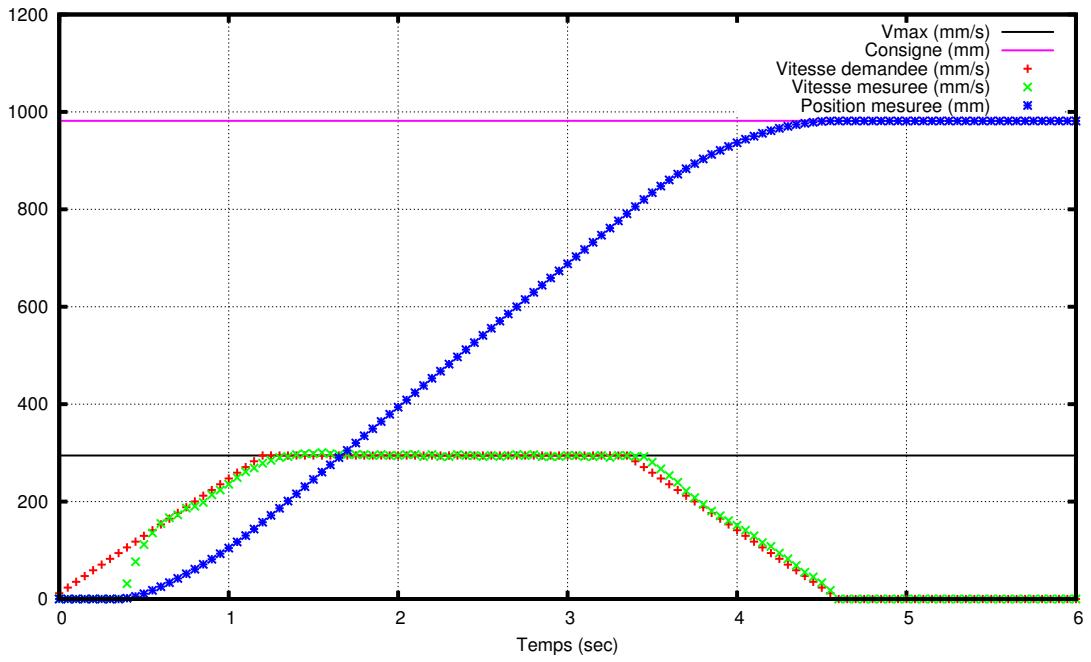
## Asservissement de position

Pour les tests, le moteur a été couplé sur une roue de 6cm de diamètre. Le réglage de l'asservissement de position consiste alors à rechercher le coefficient  $K_p$  du terme proportionnel qui engendrera la plus petite erreur statique possible sans pour autant être géné par la limitation de l'accélération maximale lors des phases de décélération. Pour les tests, les valeurs suivantes ont été choisies<sup>4</sup> :

$$\left\{ \begin{array}{ll} A_{max} = 23.5 \text{ cm.s}^{-2} & \text{Accélération maximale} \\ V_{max} = 29.5 \text{ cm.s}^{-1} & \text{Vitesse maximale} \end{array} \right.$$

On obtient alors le résultat voulu pour  $K_p = 0.08$ . L'erreur statique reste heureusement très satisfaisante : inférieure à 1mm quelle que soit la consigne (par exemple 0.3mm pour une consigne de 1m, soit 0.03% d'écart relatif). Voici la courbe de réponse mesurée pour une consigne d'environ 1m :

<sup>4</sup>Si ces valeurs semblent étonnantes, il s'agissait en réalité de valeurs brutes « rondes » faciles à utiliser pour le micro-contrôleur.



**Remarque** On s'aperçoit ici encore que les frottements secs gênent la poursuite de la consigne de vitesse au démarrage, lorsque le couple moteur  $c_m$  est encore inférieur au couple d'arrachement  $C_{arr}$ .

# Chapitre 3

## Application : Asservissement en position et en vitesse d'un robot

### 3.1 Objectif de l'asservissement

L'expérimentation consiste à réguler en vitesse et en position deux moteurs électriques à courant continu, placés en configuration « type char » sur un robot (un moteur par roue, une roue de chaque côté), lui permettant d'avancer et de tourner sur lui-même. Cette configuration est notamment utilisée dans les convoyeurs automatisés (« fenwicks » autonomes).



FIG. 3.1 –

(a) Les convoyeurs automatisés AGV utilisent une configuration "type char" des roues pour une bonne mobilité

Le besoin d'asservissement est double : d'une part, on souhaite pouvoir asservir la position du robot en absolu, c'est-à-dire savoir à tout instant où se situe le robot et pouvoir se déplacer vers des coordonnées précises. D'autre part, de nombreux éléments (légères différences de moteurs, décharge de la batterie, parasites, mauvais équilibrage du robot, ...) font que deux moteurs supposés identiques alimentés de la même manière ne réagiront pas strictement de la même manière sans asservissement, ce qui aura pour effet de faire dévier le robot au lieu d'avancer droit. Un asservissement est donc nécessaire.

### 3.2 Modélisation d'un robot à deux moteurs

Pour pouvoir modéliser l'asservissement d'un tel robot en position, il ne suffit pas seulement de modéliser le comportement des moteurs, mais il faut aussi pouvoir connaître la position et la vitesse du robot (selon une composante de translation et une composante de rotation) en fonction de l'historique d'évolution des vitesses respectives de ses moteurs. Pour cela, on met en place un référentiel mobile polaire au centre du robot.

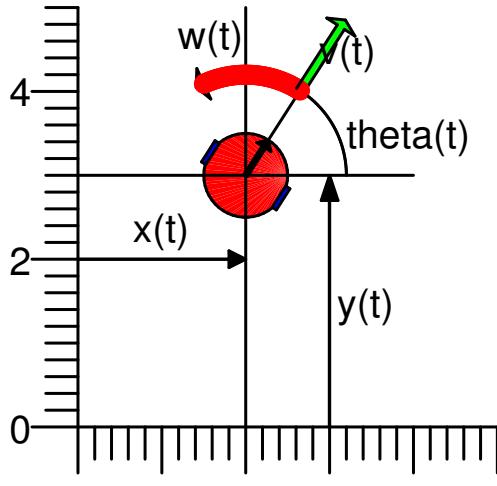


FIG. 3.2 -

(a) Modélisation d'un robot en configuration "type char"

On peut alors définir les composantes instantanées de vitesse de translation  $v$  et de rotation  $\omega$  en fonction des vitesses d'avance respectives  $V_{droite}$  et  $V_{gauche}$  des roues droite et gauche (on suppose l'environnement plat), ainsi que  $R$  la distance entre une roue et le centre du robot. On a alors :

$$\begin{cases} v = \frac{V_{droite} + V_{gauche}}{2} \\ \omega = \frac{V_{droite} - V_{gauche}}{2R} \end{cases}$$

On en déduit alors les coordonnées du robot dans le référentiel fixe, en supposant que le robot soit à l'origine du repère à l'instant initial :

$$\begin{cases} x(t) = \int_0^t v(\tau) \cdot \cos(\theta(\tau)) d\tau = \int_0^t v(\tau) \cdot \cos\left(\int_0^\tau \omega(\alpha) d\alpha\right) d\tau \\ y(t) = \int_0^t v(\tau) \cdot \sin(\theta(\tau)) d\tau = \int_0^t v(\tau) \cdot \sin\left(\int_0^\tau \omega(\alpha) d\alpha\right) d\tau \\ \theta(t) = \int_0^t \omega(\alpha) d\alpha \end{cases}$$

**Remarques** Cette modélisation suppose une symétrie parfaite des roues et un contact ponctuel et parfait entre les roues et le sol (pas de glissement, roue suffisamment fine pour connaître avec exactitude la position du centre de rotation, ...).

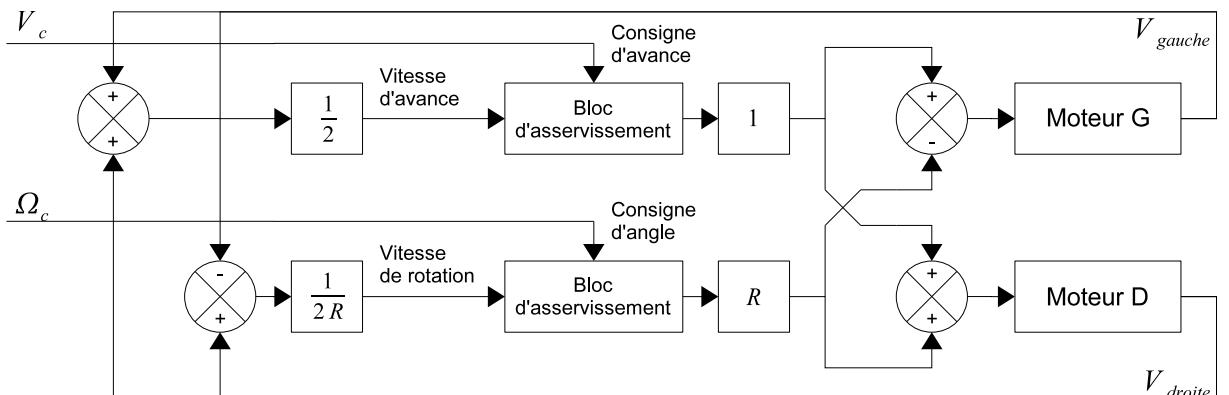


FIG. 3.3 -

(b) Schéma-bloc global de l'asservissement bimoteurs

### 3.3 Mise en oeuvre

#### 3.3.1 Partie mécanique

Le robot fabriqué pour l'occasion est mû par deux moteurs électriques à courant continu (identiques à celui étudié au chapitre précédent) dotés de réducteurs de rapport de réduction  $r = \frac{1}{30}$ . Pour coller avec le modèle

géométrique du robot, les roues doivent être fines et d'espacement connu, ce qui permet d'avoir un contact quasi-ponctuel avec le sol et de bien connaître le centre instantané de rotation (ce qui est plus aléatoire avec de larges roues).

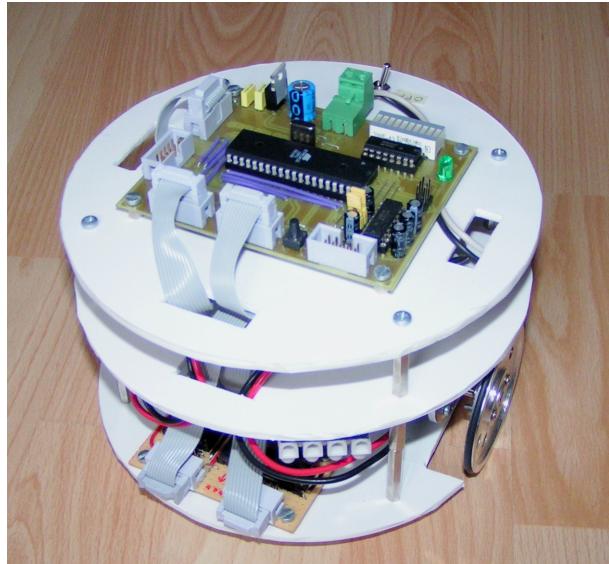


FIG. 3.4 –

(a) Vue globale de la base roulante

Pour simplifier la modélisation, on considère que le comportement global du robot dépend uniquement du comportement des moteurs, en négligeant le frottement et le jeu des réducteurs, le glissement des roues, en supposant parfaits l'alignement et la géométrie des roues, ...

Un encodeur optique incrémental à quadrature de phase identique à celui utilisé dans la précédente application est relié au rotor de chacun des deux moteurs, ce qui nous permet de récupérer une information sur l'incrémentation angulaire des moteurs, à partir de laquelle on peut calculer la vitesse du robot et sa position d'après les formules énoncées précédemment.

### 3.3.2 Partie électronique

Deux cartes électroniques assurent le bon fonctionnement du robot :

- La carte principale s'occupant de l'acquisition et du traitement des signaux est identique à celle utilisée au chapitre précédent :

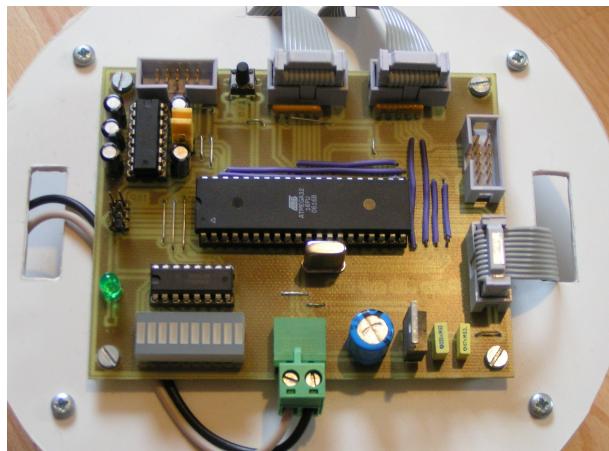


FIG. 3.5 –

(a) Carte principale

- Une carte puissance, identique à celle utilisée au chapitre précédent, permet de contrôler la tension à fournir aux moteurs (par une modulation PWM) ainsi que le sens de déplacement selon les ordres de la carte principale.

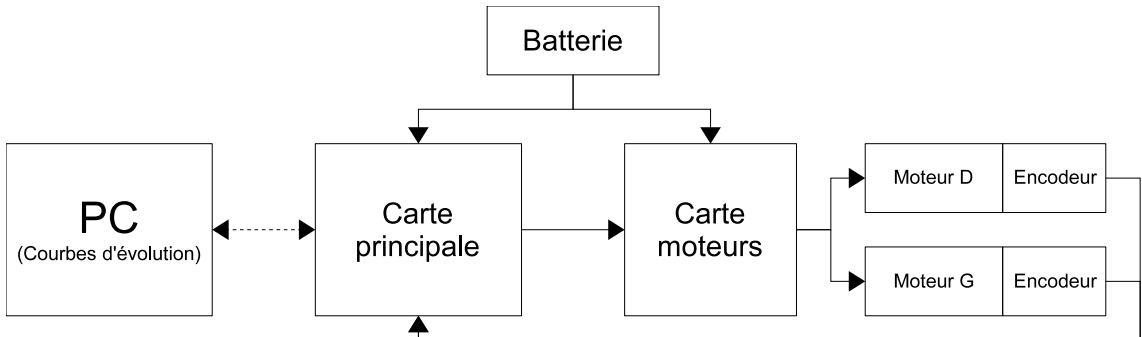


FIG. 3.6 –

(a) Schéma de fonctionnement général du robot

### 3.3.3 Partie informatique embarquée

La partie informatique est similaire à celle utilisée dans l'exemple d'application précédent. Le code a simplement été doublé de manière à calculer deux corrections simultanément. La seule particularité par rapport à la première application est que l'erreur se calcule différemment puisque l'on n'asservit pas les mêmes variables.

**Remarques** Cette implantation suppose une symétrie parfaite du robot (parallelisme des roues, ...), un contact parfait et ponctuel entre chaque roue et le sol, des roues infiniment fines, un sol parfaitement plat, ce qui bien sûr n'est qu'une approximation de la réalité. Il en résulte alors une déviation progressive du robot par rapport à la position qu'il pense tenir. Une solution serait d'utiliser d'autres capteurs en complément qui permettraient de recaler régulièrement la position du robot par rapport à un point de position connue.

### 3.3.4 Partie informatique distante

Pour la récupération des mesures et le tracé des courbes, la même méthode qu'au chapitre précédent a été utilisée.

## 3.4 Mise au point de l'asservissement - Résultats

La liaison série disponible sur la carte principale permet de récupérer les courbes d'évolution de la vitesse, nous permettant ainsi d'apprécier visuellement les effets des coefficients. Le réglage trouvé au chapitre précédent pour les mêmes moteurs a donné de bons résultats et a été utilisé aussi bien pour l'asservissement d'avance que l'asservissement de rotation.

Une fois l'asservissement au point, il est possible de demander l'exécution de trajectoires complexes en approximant chaque petit déplacement par une portion de cercle dont il reste à déterminer le rayon de courbure et la vitesse. Lors des tests, le robot a très bien « collé » à sa trajectoire sans en dévier de plus de quelques millimètres.

Pour un positionnement absolu irréprochable, il serait alors envisageable d'utiliser en complément d'autres capteurs (analysant la position par rapport à une balise par exemple) ou de se recaler de temps à autres en un point de position connue afin de limiter au maximum la dérive qui sans cela ne pourra aller qu'en augmentant.

# Chapitre 4

## Conclusion

### 4.1 Synthèse

Nous avons vu que le PID appliqué aux moteurs électriques à courant continu était un régulateur qui pouvait offrir de bonnes performances, que ce soit au niveau de la précision, du temps de réaction, du dépassement, ou de la stabilité. Diverses méthodes permettent d'optimiser les coefficients du PID. Mais le régulateur idéal n'existe pas : chaque optimisation d'un paramètre (précision, ...) se fait aux dépends d'un autre. Il convient alors de trouver le meilleur compromis en fonction des exigences initiales du cahier des charges.

Divers éléments, tels que le choix de la méthode d'implantation du PID, peuvent aussi avoir des conséquences sur les performances. L'essentiel est alors de s'éloigner le moins possible du modèle théorique pour qu'il reste valable.

Lorsque des processus industriels mettent en oeuvre un ou plusieurs moteurs, il peut être avantageux de mettre en place plusieurs régulations combinées plutôt qu'une seule, pour mettre en place un asservissement à profil de vitesse ou un asservissement multi-effecteurs.

### 4.2 Alternatives à l'asservissement PID

Il arrive dans certaines situations que le PID ne donne pas de bons résultats. Plusieurs autres actuateurs et systèmes de régulations existent pouvant donner des résultats différents, parmi lesquels :

#### Correcteur RST

Le correcteur RST est une fonction classique qui est généralement mise en place lorsqu'un PID ne donne pas satisfaction (car plus complexe et plus lourde à mettre en oeuvre). Là où le PID est avant tout destiné à un traitement analogique, le correcteur RST est une correction essentiellement numérique qui laisse le choix de la fonction de correction sous la forme d'un quotient polynomial, ce qui permet d'imposer certains pôles mais aussi certains zéros en boucle fermée. Il autorise alors la poursuite sans écart permanent de consignes quasi-quelconques.

#### Réseaux Neuronaux Artificiels

Là où l'algorithmique traditionnelle trouve ses limites (reconnaissance de formes, asservissement de systèmes complexes, ...), les réseaux neuronaux artificiels offrent une alternative de plus en plus appréciée : si un système biologique arrive à exécuter une certaine tâche, alors on tente d'émuler les réseaux neuronaux naturels pour s'approprier leurs incroyables propriétés : exécution de tâches très complexes, auto-correction des erreurs, faculté d'auto-apprentissage, bonne improvisation face à l'inconnu, incroyable résistance aux perturbations et aux casses matérielles.

- Certains des récents robots humanoïdes nippons s'en servent par exemple pour maintenir leur équilibre dynamiquement et pour reconnaître des visages. Le maintien de l'équilibre dynamique d'un humanoïde est pourtant une tâche très complexe et agissant sur de nombreux effecteurs (hanches, genoux, ...) qu'il est très difficile de réaliser par une approche algorithmique classique.
- Mark Tilden (personne très reconnue en robotique) a créé des robots d'une intelligence comparable aux insectes, capables d'improviser leur « survie » malgré une patte cassée, ce qui montre bien leur robustesse.

#### Autres types de moteurs

Les moteurs pas à pas (qui équipent de nombreuses machines outils à commande numérique, télescopes robotisés, tables traçantes, imprimantes matricielles, ...) et les moteurs piézoélectriques (micro-robotique, autofocus d'appareils photo, ...) ont la particularité de ne tourner que par pas réguliers (par exemple 200 pas = 1 tour).

Ainsi il est possible (lorsque le couple opposé reste suffisamment faible pour que le moteur tourne) d'« espérer » la position angulaire du moteur en fonction du nombre de pas demandés (mais il ne s'agit alors pas réellement d'un asservissement, mais juste d'une supposition de la position).