

## Introduction

Le but de ce document est de rassembler les informations nécessaires au développement du code informatique de la partie *Déplacement* du projet intégré.

La première partie de ce document contient les informations techniques concernant les périphériques d'interface entre le microcontrôleur et le robot.

La seconde partie décrit le développement du régulateur de position à implémenter.

## Informations techniques

### Le dsPIC Déplacement

Le dsPIC *Déplacement* est un dsPIC33FJ128MC804. C'est un processeur spécialisé dans le contrôle de moteur. Il possède 2 périphériques d'interface pour encodeur en quadrature et un périphérique *Output Compare* capable de générer 4 PWM, soit tout ce dont nous avons besoin pour contrôler simultanément 2 moteurs.

**Remarque** : le dsPIC33FJ128MC804 possède également un périphérique appelé *motor control PWM module*, qui permet de générer des formes de PWM complexes. N'ayant besoin que d'une "simple" PWM par moteur, nous utiliserons le *Output Compare*, plus simple à utiliser.

Il possède aussi deux périphériques d'interface UART que nous utiliserons pour la communication entre les deux processeurs.

### Interface avec les moteurs

Le robot est équipé de servomoteurs commandés en tension. Un servomoteur est un système intégrant un moteur à courant continu, un réducteur pour obtenir une vitesse plus faible (et un couple plus important) à l'axe de sortie, un hacheur pour alimenter le moteur et un circuit de contrôle qui commande le hacheur.

Les servomoteurs ont 3 fils d'interface : une masse, une alimentation et un signal de commande.

Pour contrôler le moteur, le dsPIC devra envoyer une consigne au circuit de contrôle du moteur, en appliquant le signal de commande suivant :

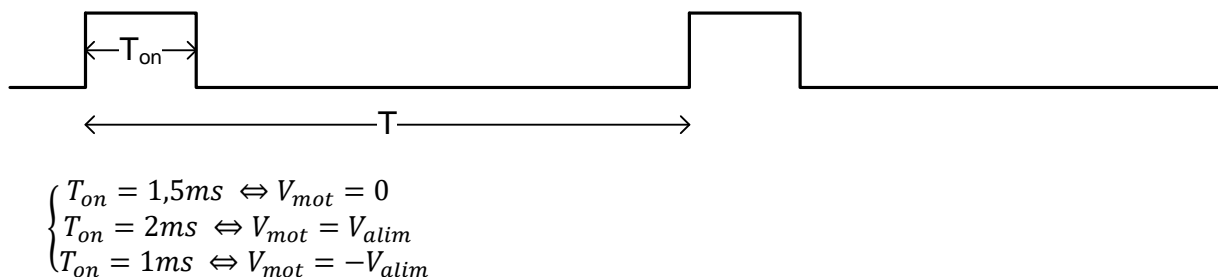


Figure 1 : Signal de commande des servomoteurs

La période  $T$  du signal n'a pas d'importance, elle doit simplement être comprise entre 3ms et 20ms.

La consigne est codée dans la durée de l'impulsion  $T_{on}$  ; elle définit la tension qui sera appliquée au moteur suivant la relation donnée ci-dessus.

### Principe de l'encodeur en quadrature

Dans les applications de robotique ou de motorisation, il arrive fréquemment que l'on souhaite mesurer le déplacement d'une roue. Par exemple, cela permet de connaître le nombre de tours parcourus par les roues d'une voiture et de suivre son mouvement à des fins de régulation et/ou de contrôle.

Le capteur de rotation peut prendre de nombreuses formes, mais une des plus utilisées est l'encodeur incrémental en quadrature. Ce dispositif est composé d'une roue perforée solidaire de l'axe en rotation, et d'un capteur (souvent optique) permettant de détecter le passage d'une fente de la roue. Un second capteur, décalé du premier, permet de connaître le sens de déplacement.

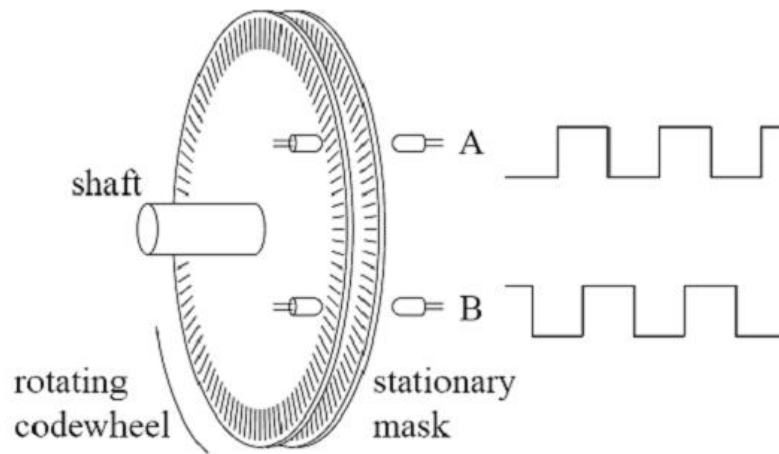


Figure 2 : Principe d'un encodeur en quadrature

Le passage de la roue génère deux trains d'ondes carrées, nommés A et B. Selon que le flanc montant de A arrive avant ou après le flanc montant de B, on sait dans quel sens se déplace la roue. Chaque flanc correspond donc à la rotation d'une fraction connue de la roue dans le sens positif ou négatif.

Le fonctionnement des encodeurs du robot est décrit dans le document *VEX - Quadrature encoder*. Chaque canal fournit 90 impulsions par tour. En utilisant le mode 4x du périphérique QEI, on peut obtenir 360 impulsions par tour. Les encodeurs sont fixés aux axes des roues.

#### Interface avec les encodeurs en quadrature

Le dsPIC33FJ128MC804 possède 2 périphériques conçus pour interpréter les signaux d'un encodeur en quadrature : les *Quadrature Encoder Interface* (QEI).

Le rôle du QEI est de surveiller l'évolution de ces deux signaux et d'incrémenter (ou de décrémenter selon le sens de rotation) un compteur. La lecture de ce compteur permet de surveiller l'évolution du déplacement. Leur fonctionnement est décrit dans le document *Programmation d'une carte à  $\mu C$* .

#### Description du régulateur utilisé

Nous allons utiliser la régulation polaire, classiquement utilisée pour le positionnement des robots.

Elle permet de contrôler séparément les mouvements de translation et de rotation.

Nous allons d'abord étudier le cas de la régulation d'un seul moteur.

#### Régulateur de position pour un moteur

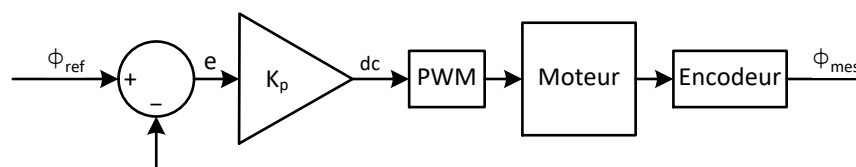


Figure 3 : Schéma du régulateur de position

La Figure 3 montre un régulateur de position pour un seul moteur :

- Au centre, nous avons le bloc *Moteur* qui représente la fonction de transfert du moteur et du système mécanique qu'il actionne.
- Il est suivi du bloc *Encodeur* qui fournit la mesure de la position :  $\Phi_{mes}$ . Ici, il s'agit de la position angulaire du moteur.
- L'erreur de position  $e$  est calculée en soustrayant  $\Phi_{mes}$  à la position désirée  $\Phi_{ref}$ .
- Nous nous contenterons d'un régulateur proportionnel de gain  $K_p$ , qui fournit le signal de commande à appliquer au moteur. Classiquement, on exprime la commande à appliquer au moteur par le rapport cyclique (ou Duty Cycle)  $dc$ . Il représente la fraction de la tension d'alimentation (ici la batterie 7,2V) à appliquer au moteur. ; Il peut donc varier entre -1 et 1 (correspondant à  $-V_{alim}$  et  $+V_{alim}$ ).

- Enfin, le bloc *PWM* transforme le rapport cyclique en un signal de commande adapté au moteur utilisé.

Nous pouvons modifier ce régulateur pour contrôler les mouvements de translation ou de rotation de notre robot. Nous allons dans un premier temps étudier les deux cas séparément.

### Régulateur de translation

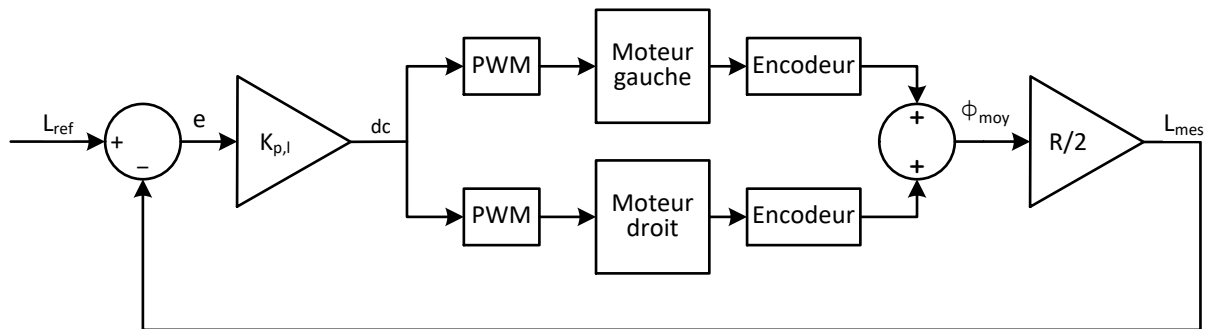


Figure 4 : Schéma du régulateur de translation

Dans le cas d'une translation, les deux roues doivent parcourir la même distance, en même temps.

- Nous appliquons donc le même signal de commande aux deux moteurs.
- Pour la mesure de position, nous utilisons la moyenne des mesures des deux encodeurs  $\Phi_{moy}$ .
- La grandeur que nous voulons contrôler est la distance parcourue par le robot, nous multiplions donc la position angulaire par le rayon des roues ( $R$ ).
- Ce régulateur contrôle la distance parcourue par le robot, mais si les deux moteurs n'ont pas exactement la même fonction de transfert, la trajectoire du robot sera un arc de cercle et non une droite.

### Régulateur de rotation

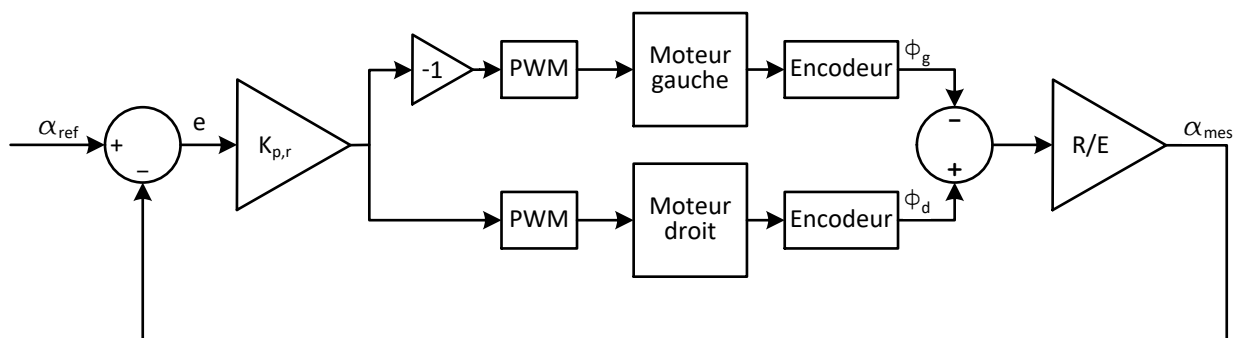


Figure 5 : Schéma du régulateur de rotation

Dans le cas d'une rotation, les deux roues doivent parcourir la même distance, mais dans des sens opposés. Le point situé au milieu des deux roues est le centre de la rotation.

- On choisit le sens trigonométrique comme sens positif. La roue droite tourne alors vers l'avant et la roue gauche vers l'arrière. On doit donc inverser le signe de la commande appliquée au moteur gauche.
- On veut contrôler la position angulaire du robot  $\alpha$ . Un petit calcul géométrique nous donne  $\alpha = \frac{R}{E} \cdot (\Phi_d - \Phi_g)$ , où  $E$  est l'empattement du robot (la distance entre les deux roues).
- Ce régulateur contrôle l'angle parcouru par le robot, mais si les deux moteurs n'ont pas exactement la même fonction de transfert, le milieu des roues se déplacera.

### Combinaison des deux régulateurs

La Figure 6 montre le régulateur polaire qui est la combinaison des deux précédents.

- Pour une translation, il suffit de changer la consigne de distance  $L_{ref}$  en gardant la consigne d'angle  $\alpha_{ref}$  constante.  
Le régulateur de translation agit comme précédemment, en agissant sur la distance moyenne parcourue par les deux roues.  
Le régulateur de rotation agit pour compenser une éventuelle différence entre les deux moteurs en gardant les distances parcourues par les deux roues égales.
- On peut faire le même raisonnement pour une rotation en inversant les rôles des régulateurs.

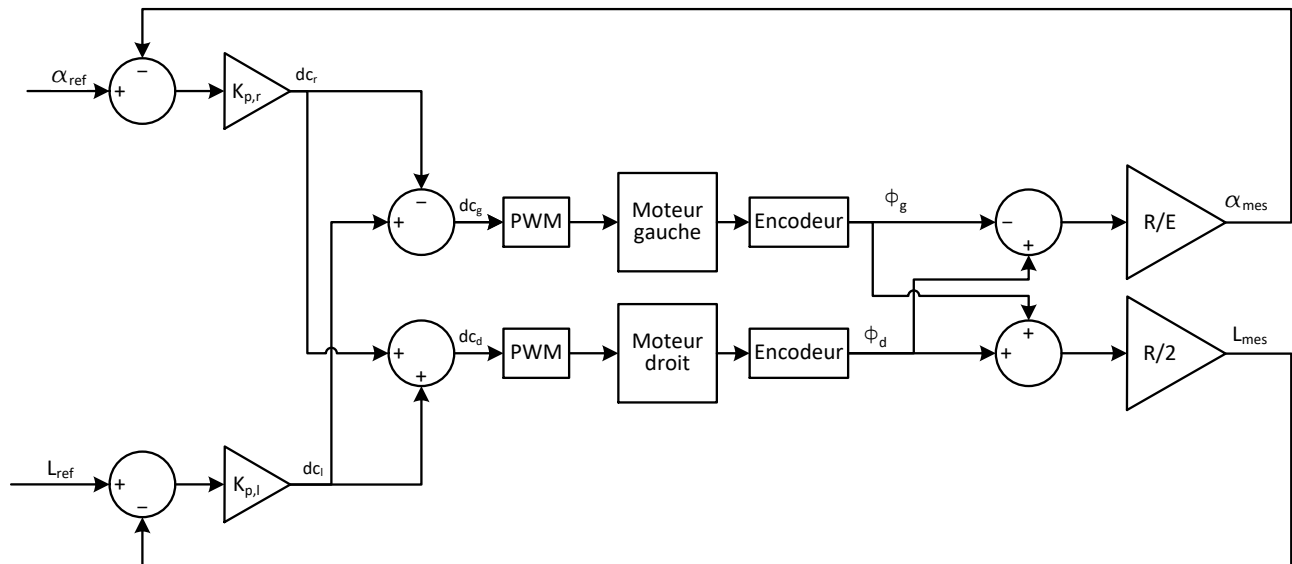


Figure 6 : Schéma du régulateur polaire

### Dimensionnement du régulateur de translation

Pour dimensionner les régulateurs, nous devons obtenir un modèle du robot. Cette section décrit en détail le dimensionnement du régulateur de translation. La même méthode peut être appliquée pour le régulateur de rotation.

#### Modélisation du robot

##### Introduction

Comme nous n'avons pas d'informations sur le fonctionnement interne des moteurs, nous allons utiliser une approche empirique pour créer un modèle de notre robot.

Chaque roue du robot est équipée d'un encodeur en quadrature possédant 90 dents, ce qui nous permet de mesurer la position angulaire des roues avec une précision de  $1^\circ$ . Les roues font 4" de diamètre (101,6mm).

##### Mesures

Nous savons qu'il existe une relation entre la tension appliquée au moteur (définie par le rapport cyclique) et sa vitesse. Pour obtenir cette relation, nous allons relever la réponse de la vitesse du robot à un échelon de tension de 0 à la valeur désirée. Cet échelon est appliqué aux deux moteurs simultanément, pour obtenir un mouvement de translation du robot.

Nous allons travailler avec une fréquence d'échantillonnage de 100Hz, qui est une valeur courante pour le réglage des moteurs en robotique. La vitesse est estimée en comptant le nombre d'impulsions durant une période d'échantillonnage.

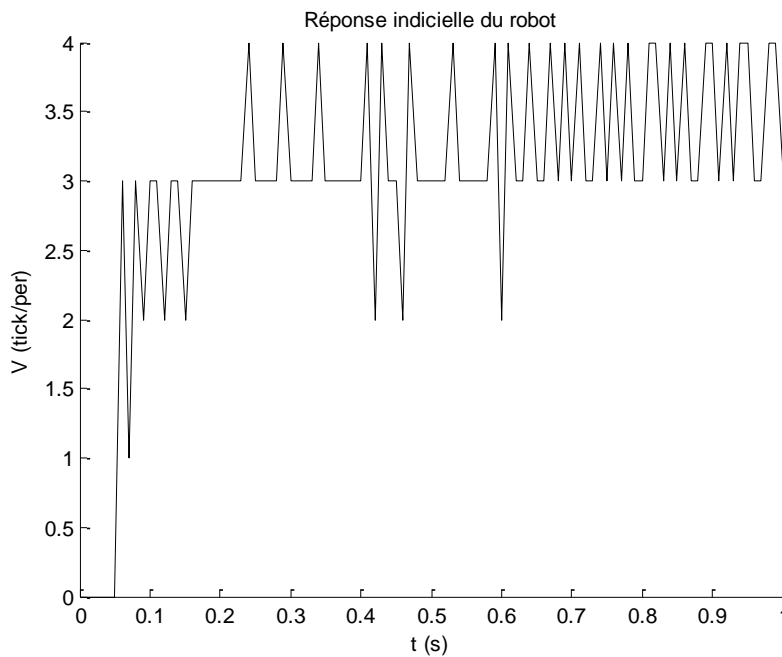


Figure 7 : Réponse indicielle du robot (mesures brutes)

La Figure 7 donne un exemple de la réponse obtenue pour une des roues en utilisant les mesures brutes. On voit qu'elle est fortement bruitée par la quantification.

Pour rendre ces courbes plus lisibles, on fait la moyenne des courbes des deux roues et on la filtre avec un filtre passe-bas d'ordre 2 dont la fréquence de coupure est 20Hz.

La Figure 8 montre les courbes ainsi obtenues pour plusieurs tensions. On observe un transitoire qui peut être approché par un système du 1<sup>er</sup> ordre, suivi d'une phase de régime où la vitesse est relativement constante. L'ondulation périodique qu'on observe est probablement due à une variation dans les frottements en fonction de la position angulaire de la roue (la période est compatible avec le diamètre de la roue et la vitesse).

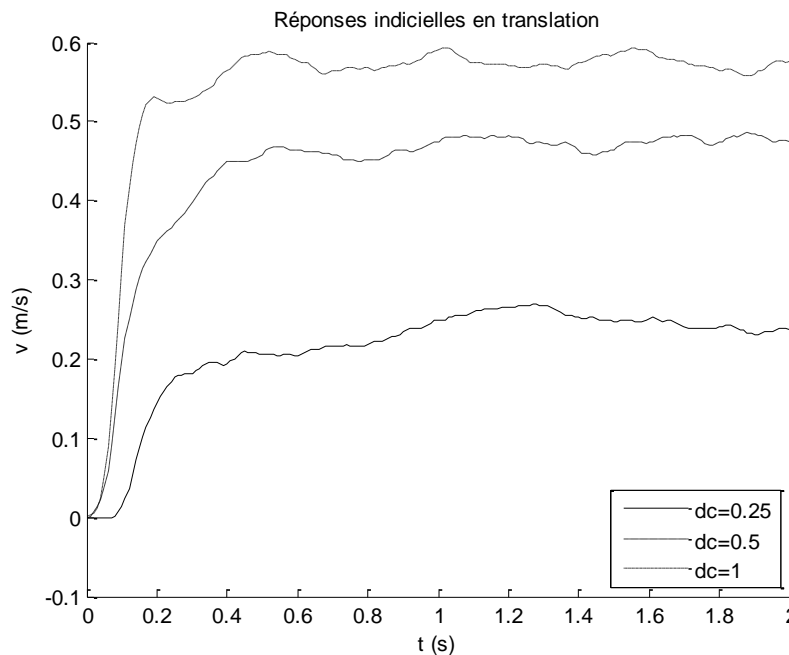


Figure 8 : Réponses indicielles filtrées

### Caractéristique statique

Pour obtenir la caractéristique statique de la vitesse du robot en translation, nous avons calculé la moyenne de la vitesse en régime, par pas de 5% de rapport cyclique (défini comme le rapport entre la tension appliquée au moteur et celle de la batterie, il est donc compris entre 0 et 1).

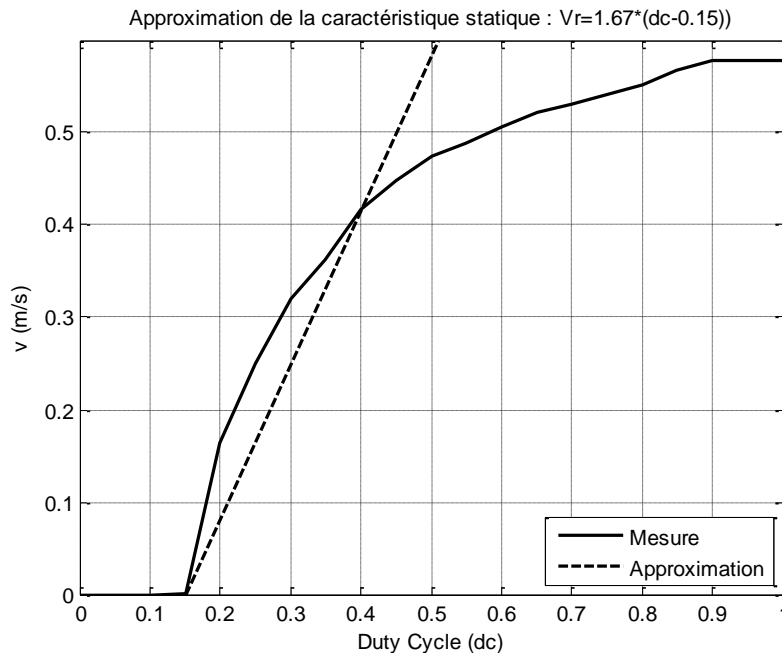


Figure 9 : Caractéristique statique en vitesse

Nous obtenons une courbe (alors que nous nous attendions à avoir une droite). Comme nous voulons un modèle linéaire de notre robot, nous allons approximer cette courbe en choisissant une vitesse nominale pour notre robot. Nous choisissons 0,4m/s pour que notre droite ne s'éloigne pas trop de la courbe.

D'autre part, nous devons tenir compte dans notre modèle de la zone morte pour les rapports cycliques en dessous de 0,15. En effet, dans cette zone, le régulateur n'a aucun effet sur le robot, il faut donc à tout prix l'éviter.

Nous obtenons alors une caractéristique linéaire par morceau :

$$\begin{cases} V_r = k_v \cdot (\delta + \delta_0) & \text{si } \delta < -\delta_0 \\ V_r = 0 & \text{si } -\delta_0 \leq \delta \leq \delta_0, \\ V_r = k_v \cdot (\delta - \delta_0) & \text{si } \delta > \delta_0 \end{cases} \quad \text{avec } k_v = 1,67 \text{ m/s} \text{ et } \delta_0 = 0,15$$

Cette approximation s'éloigne fortement de la courbe expérimentale pour des vitesses au-delà de la vitesse nominale ; mais nous ne comptons pas utiliser cette partie de la courbe de toute façon.

### Modèle dynamique

Pour modéliser le comportement dynamique du robot, nous allons utiliser la réponse indicielle à la vitesse nominale et l'interpoler par une réponse indicielle d'un système du 1<sup>er</sup> ordre :

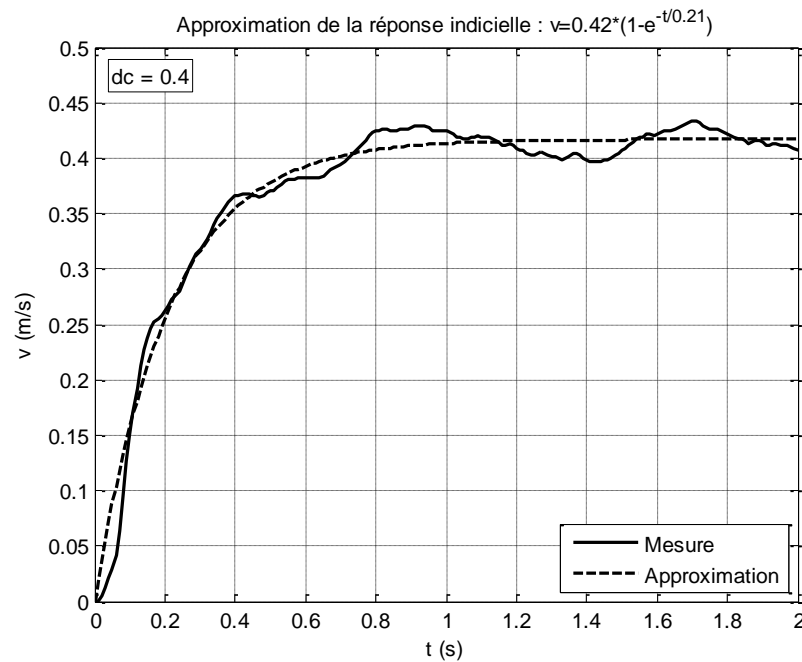


Figure 10 : Approximation de la réponse indicielle à la vitesse nominale

Nous obtenons une constante de temps :  $\tau = 210\text{ms}$  ; le gain correspond évidemment à la vitesse nominale.

### Modèle complet

Nous avons maintenant tous les blocs nécessaires pour modéliser le comportement de notre robot en translation :

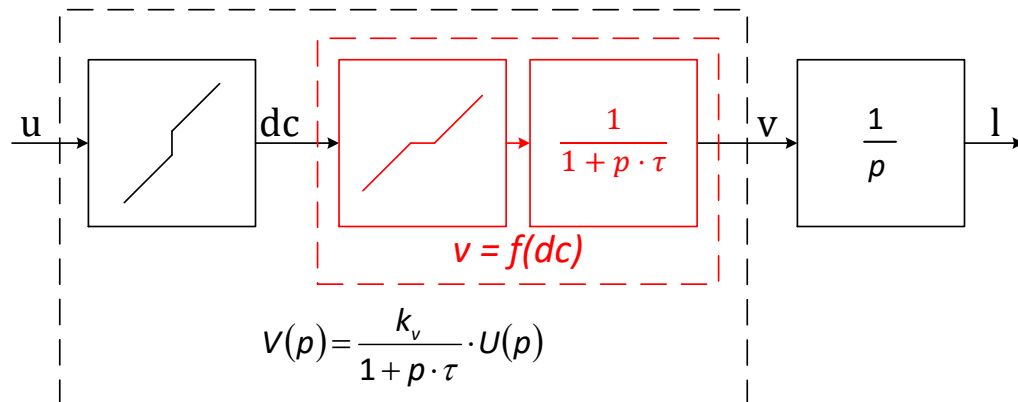


Figure 11 : Modèle du robot en translation

Les blocs rouges représentent respectivement la caractéristique statique et la réponse indicielle. Ils représentent la fonction de transfert en vitesse de notre robot. Malheureusement, elle n'est pas linéaire à cause de la zone morte de la caractéristique statique.

Pour rendre notre système linéaire, nous allons ajouter une non-linéarité de commande pour transformer notre système  $V = f(\delta)$  en un système  $V = f'(u)$  linéaire. Ce bloc est décrit par :

$$\delta = \begin{cases} u - \delta_0 & \text{si } u < 0 \\ 0 & \text{si } u = 0 \\ u + \delta_0 & \text{si } u > 0 \end{cases}$$

Comme nous voulons une régulation en position, il nous suffit alors d'ajouter un intégrateur pour obtenir notre fonction de transfert en position :

$$H(p) = \frac{L(p)}{U(p)} = \frac{k_v}{(1 + p \cdot \tau) \cdot p}$$

**Schéma de régulation**

Notre système possède un pôle intégrateur, nous pouvons donc nous contenter d'un simple régulateur proportionnel :

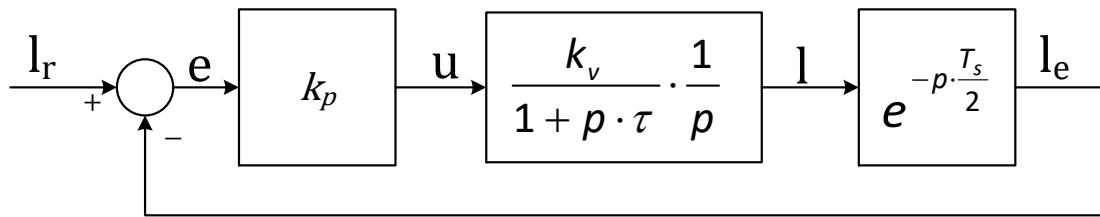


Figure 12 : Schéma de régulation

Le bloc le plus à droite représente l'échantillonnage de la mesure de vitesse. C'est un délai d'une demi-période d'échantillonnage.

Nous obtenons la fonction de transfert en boucle ouverte de notre système :

$$BO(p) = \frac{L_e(p)}{E(p)} = \frac{k_p \cdot k_v \cdot e^{-p \cdot \frac{T_s}{2}}}{(1 + p \cdot \tau) \cdot p}$$

**Dimensionnement du régulateur**

Nous allons utiliser les critères des marges de gain et de phase pour dimensionner notre régulateur.

Nous avons donc besoin des expressions du gain et de la phase de la réponse harmonique du système en boucle ouverte :

$$\begin{cases} \|BO(\omega)\| = \frac{k_p \cdot k_v}{\omega \cdot \sqrt{1 + (\omega \cdot \tau)^2}} \\ \phi(BO(\omega)) = -\omega \cdot \frac{T_s}{2} - \arctg(\omega \cdot \tau) - \frac{\pi}{2} \end{cases}$$

Nous voulons une marge de gain d'au moins 6dB et une marge de phase d'au moins 30°. Ce sont des valeurs de bonnes pratiques couramment utilisées.

**Marge de gain**

La marge de gain est définie comme l'inverse du module de la réponse harmonique en boucle ouverte à la pulsation pour laquelle sa phase vaut -180°. Cette marge de gain doit être supérieure à 1 pour que le système soit stable.

Soit  $\omega_1$  la pulsation recherchée :

$$\phi(BO(j\omega_1)) = -\pi \Leftrightarrow \arctg(\omega_1 \tau) + \omega_1 \frac{T_s}{2} = \frac{\pi}{2}$$

En résolvant cette relation numériquement, on trouve  $\omega_1 = 30,7 \text{ rad}$

Le critère de la marge de gain peut s'écrire :

$$\begin{aligned} \|BO(j\omega_1)\| = -6\text{dB} &= \frac{1}{2} = \frac{k_{p1} \cdot k_v}{\omega_1 \cdot \sqrt{1 + (\omega_1 \tau)^2}} \\ \Rightarrow k_{p1} &= \frac{\omega_1 \cdot \sqrt{1 + (\omega_1 \tau)^2}}{2 \cdot k_v} = 60,2 \text{ m}^{-1} \end{aligned}$$

**Marge de phase**

La marge de phase est définie comme la différence entre la phase de la réponse harmonique en boucle ouverte à la pulsation pour laquelle son module vaut 1 et -180°. Cette marge de phase doit être positive pour que le système soit stable.

Soit  $\omega_2$  la pulsation recherchée, la marge de phase peut s'écrire :

$$M_\phi = \frac{\pi}{6} = \phi(BO(j\omega_2)) - (-\pi) = -\omega_2 \frac{T_s}{2} - \arctg(\omega_2 \tau) - \frac{\pi}{2} + \pi$$



$$\Leftrightarrow \arctan(\omega_2 \tau) + \frac{T_s}{2} = \frac{\pi}{3}$$

En résolvant cette relation numériquement, on trouve  $\omega_2 = 7,54 \text{ rad}$ .

$\omega_2$  est définie par l'expression :

$$\|BO(j\omega_2)\| = 1 \Leftrightarrow \frac{k_{p1} \cdot k_v}{\omega_2 \cdot \sqrt{1 + (\omega_2 \tau)^2}} = 1$$

$$\Rightarrow k_{p2} = \frac{\omega_2 \cdot \sqrt{1 + (\omega_2 \tau)^2}}{k_v} = 8,49 \text{ m}^{-1}$$

### Conclusion

Nous choisirons évidemment le  $k_p$  le plus petit de ceux obtenus par les différents critères de dimensionnement :

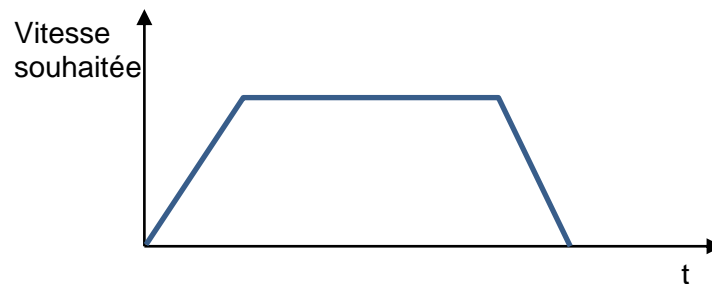
$$k_p = 8,49 \text{ m}^{-1}$$

### Génération de consigne

Pour que le déplacement du robot ne soit pas trop brutal et que les régulateurs puissent fonctionner correctement, il faut que les consignes qui leur sont fournies correspondent à une trajectoire physiquement réalisable.

Nous avons choisi une trajectoire pour laquelle l'évolution de la vitesse au cours du temps est un trapèze ; elle est donc divisée en 3 parties :

- Une phase d'accélération, où la vitesse, nulle au départ, croît linéairement jusqu'à atteindre la vitesse nominale
- Une phase à vitesse constante
- Une phase de décélération, semblable à la phase de décélération.



En intégrant ce profil de vitesse de translation/angular, nous obtenons une trajectoire de distance/orientation, qui sera la consigne du régulateur correspondant.

On peut estimer l'accélération maximale du robot en se basant sur la pente à l'origine de la réponse indicielle de la Figure 10 ; on trouve  $2 \text{ m/s}^2$ .

Après avoir essayé plusieurs accélérations, nous décidons de limiter l'accélération de notre consigne à  $0,5 \text{ m/s}^2$  pour que notre robot puisse la suivre sans difficulté.

### Modélisation du robot en rotation

Pour pouvoir dimensionner le régulateur de rotation, nous avons besoin d'un modèle. Pour cela, les mêmes expériences que pour le modèle en translation ont été effectuées. Vous trouverez les résultats ci-dessous.

Approximation de la caractéristique statique :  $V_r = 8.4 \cdot (dc - 0.25)$

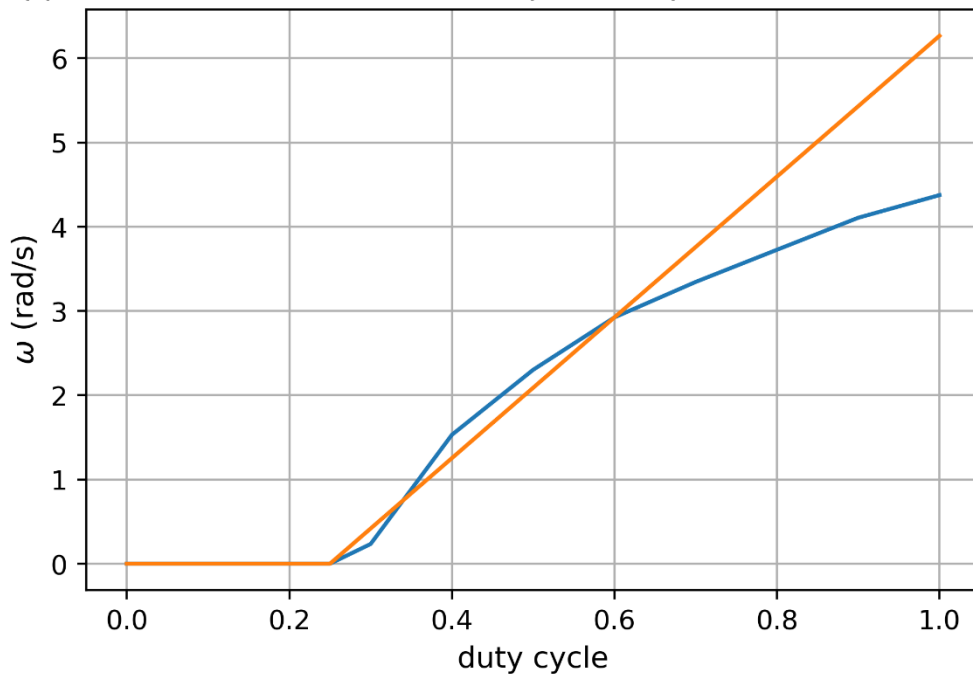


Figure 13 : Caractéristique statique en rotation

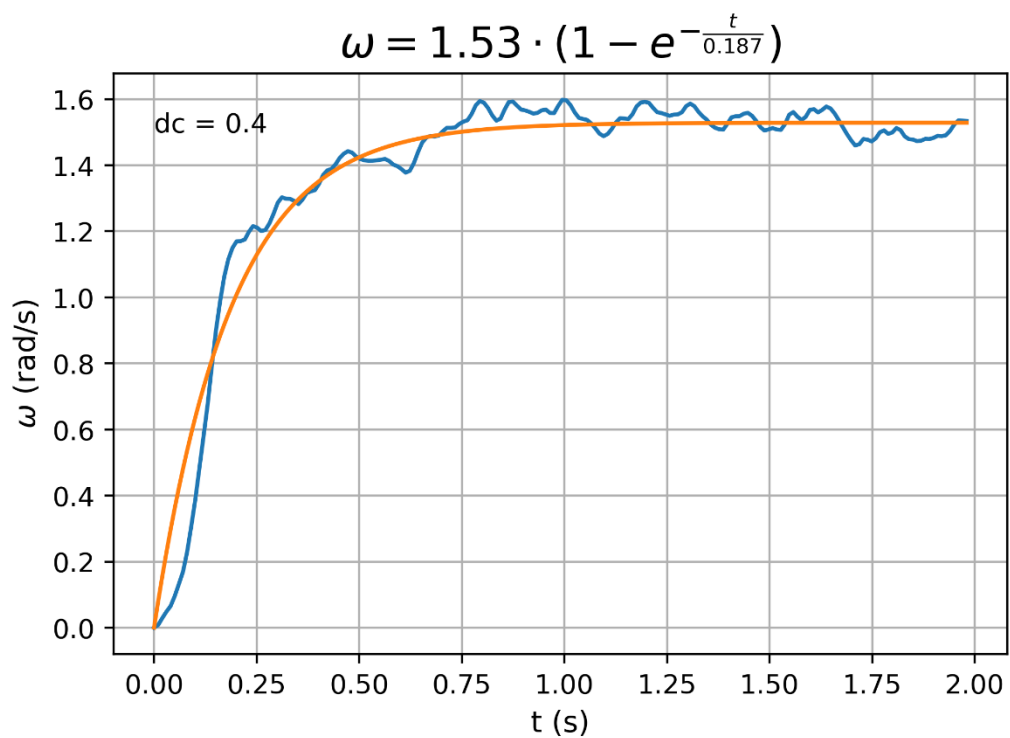


Figure 14 : Approximation de la réponse indicielle à la vitesse nominale