

1 Introduction

Le but de ce travail pratique a été de contrôler en vitesse un moteur DC. Pour cela, le système est exprimé comme une contrôleur en vitesse en cascade avec un contrôleur de courant et le moteur DC, comme illustré sur 1.

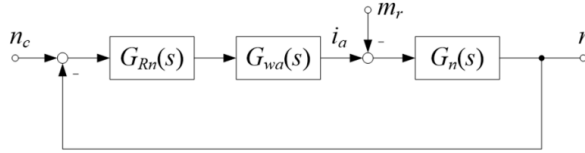


Figure 1: Block diagram du système

Le système à contrôler (le moteur DC) est approximé par la fonction de transfert:

$$G_n(s) = \frac{1}{sT_m}$$

Le contrôleur de courant est approximé par une fonction de 1er ordre:

$$G_{wa}(s) = \frac{1}{1 + sT_{ea}}$$

En prenant en compte les délais de temps ($T_E/2$, liés aux convertisseurs analogue-digital, et T_r , le retard avant que le courant ne réagisse à une commande), cette fonction de transfert devient

$$G_{pE} = \frac{1}{1 + sT_{pE}}$$

avec $T_{pe} = T_E/2 + T_{ea} + T_r$.

Pour réguler la vitesse le moteur DC, un contrôleur PI est mis en place afin de réguler la consigne en courant du banc moteur en fonction d'une consigne en vitesse. Dans la mesure où le système à contrôler est du 1er ordre, comme décrit auparavant, il est exprimé par la fonction de transfert suivante:

$$G_{Rn} = \frac{1 + sT_n}{sT_i}$$

1.1 Contrôleur

Le contrôleur PI implémenté a plusieurs modes de fonctionnement, dépendant des ajustements faits sur la consigne, l'input ou l'erreur.

1.2 Mode Standard

Dans le mode Standard, la limitation de l'input du contrôleur est implémentée: elle sert à contraindre les valeurs d'entrée à rester en-dessous du nominal .

1.3 Mode Correcteur

Dans le mode Correcteur, en plus de la limitation du input du contrôleur, le correcteur de consigne est implémenté. Ceci permet d'éviter des overshoots lors de changements soudains de consignes.

1.4 Mode Anti-Reset Windup

Dans le mode Anti-Reset Windup, la limitation de l'input du contrôleur, ainsi que l'anti-reset windup sont implémentés. Ce mode empêche le terme intégrateur du PI de croître exponentiellement, et donc que le contrôleur soit plus stable et oscille moins.

1.5 Mode Global

Le mode Global met en œuvre toutes les corrections présentes dans les modes précédents: la limitation de l'input du contrôleur, le correcteur de consigne et l'anti-reset windup.

2 Algorithme de réglage

L'algorithme de réglage consiste en un régulateur PI standard, exprimé par son équation aux différences finies:

$$y_R[k] = K_p e[k] + K_i \sum_{i=0}^k e[i]$$

où $e[k]$ est l'erreur de consigne, et où le terme $\sum_{i=0}^k e[i]$ est défini comme une variable persistente appelée **integrator** afin de garder la valeur de l'exécution précédente en mémoire.

En fonction du mode de réglage (Standard, global, etc), les termes du régulateur sont modifiés, ou limités afin d'améliorer la performance.

2.1 Correcteur de consigne

Dans les modes où le correcteur de consigne est utilisé, on calcule les coefficients $K_{wc1} = \exp(-\frac{T_E}{T_n})$ et $K_{wc1} = 1 - K_{wc1}$, et la consigne w_c est modifiée ainsi :

$$w_c[k] = K_{wc1} w_c[k] + K_{wc1} \cdot r[k]$$

avec $r[k]$ la consigne sans correction.

2.2 Limitation de la consigne

Dans le cas des modes standard, ARW et Global, on sature la consigne à une valeur absolue maximum correspondant à 2 p.u. du courant nominal de la MCC à l'aide de simples conditions **if, else**.

2.3 ARW

Dans le cas de l'ARW, on limite la consigne comme expliqué au point précédent, et on limite également l'erreur de la manière suivante:

$$e_{lim}[k] = e[k] - \frac{1}{K_{pid}}(r[k] - w_c[k])$$

avec $K_{pid} = K_p + K_i$. On utilise ensuite l'erreur limitée e_{lim} dans le calcul de la consigne y_R comme précédemment.

3 Identification des paramètres

3.1 Mesure de t_a et T_r

Le temps d'établissement du courant t_a représente entre autre le temps nécessaire pour que le courant transitoire atteigne et reste dans une bande de tolérance prédéfinie après une perturbation (comme une variation de charge).

Le temps de réaction T_r représente le retard initial avant que le courant ne commence à réagir à une sollicitation.

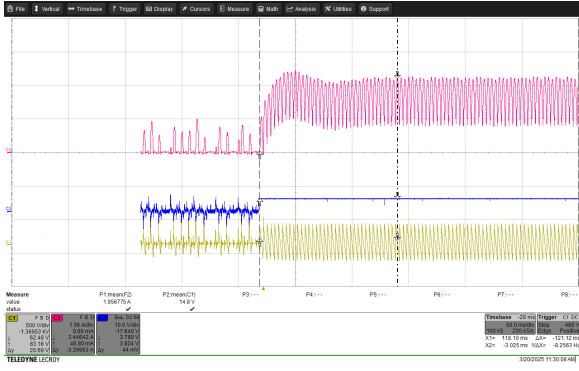
Pour identifier les paramètres t_a et T_r , on utilise un oscilloscope. Les mesures du courant et de la tension sont prises à l'armature de la machine à courant continu (MCC), avec le banc moteur en mode courant. On utilise une source DC ainsi qu'un interrupteur sans rebond comme commande en courant du banc afin de provoquer un saut de courant précis. On fixe le trigger de l'oscilloscope sur la commande venant de la source afin de pouvoir ensuite aussi analyser de manière précise le temps de réaction.

La source DC est réglée à : $3.65[V]$, ce qui correspond à une commande en courant de $\frac{I_{MCC-nominal}}{2} = \frac{7.3}{2} = 3.65[A]$. Ici on divise le courant nominal de la MCC par deux, ce qui correspondra dans notre implémentation à 1 p.u. Cela nous permettra ensuite de pouvoir dépasser 1 p.u tout en respectant les limites de la machine MCC.

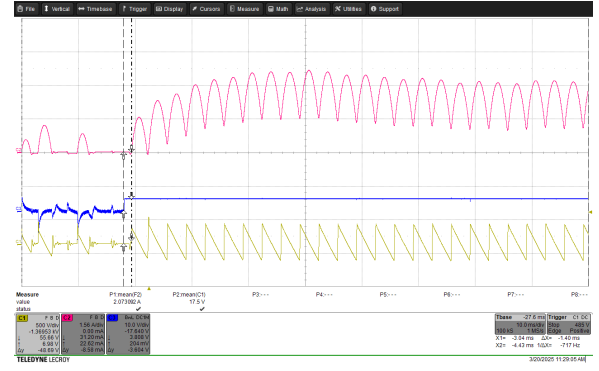
Mesures	T_r [s]	t_a [s]
1	0.0014	0.165
2	0.0020	0.167
3	0	0.169
4	0.0014	0.170
5	0	0.168
6	0.0052	0.1675
7	0.0074	0.1695
Moyenne	0.0025	0.168

Table 1: Valeurs de T_r et t_a

Avec ces mesures 1, T_{ea} peut être calculé selon : $T_{ea} = \frac{t_a}{6}$. La division par 6 est due à la conception de l'électronique de puissance de l'inverseur du banc. La valeur finale de ce paramètre (en prenant la moyenne des valeurs de t_a) est $T_{ea} = 0.028[s]$



(a) Mesure de t_a



(b) Mesure de T_r

Figure 2: Image de l'oscilloscope, courbe rose =tension ; en bleu la machine d'essai ; en jaune le courant

3.2 Mesure de T_m

Dans cette section nous mesurons T_m tel que $dn = 1 \rightarrow T_m = dt$. Attention cette relation n'est valide qu'en utilisant des valeurs en p.u et donc seulement pour la vitesse de rotation nominale de la MCC 1500 [rpm].

Son courant nominal est 7.3 A, donc l'équivalent de 2 p.u comme expliqué en section 3.1. Ainsi la limite du régulateur est donc égale à 2 pu.

Le courant d'excitation nominal de la MCC est $I_e = 0.45[A]$, tandis que l'armature est réglées à une tension de 3.65 V toujours à l'aide de la machine DC (comme précédemment).

On branche l'oscilloscope comme précédemment en mesurant aussi la vitesse de rotation n [rpm], ici une division (en hauteur) correspond à 5V et donc 1500 rpm (la vitesse nominale de la MCC). Cette mesure correspond à la courbe verte 3.

Mesures	T_m (s)
1	2.00
2	1.70
3	1.88
Moyenne	1.86

Table 2: Tableau des valeurs de T_m

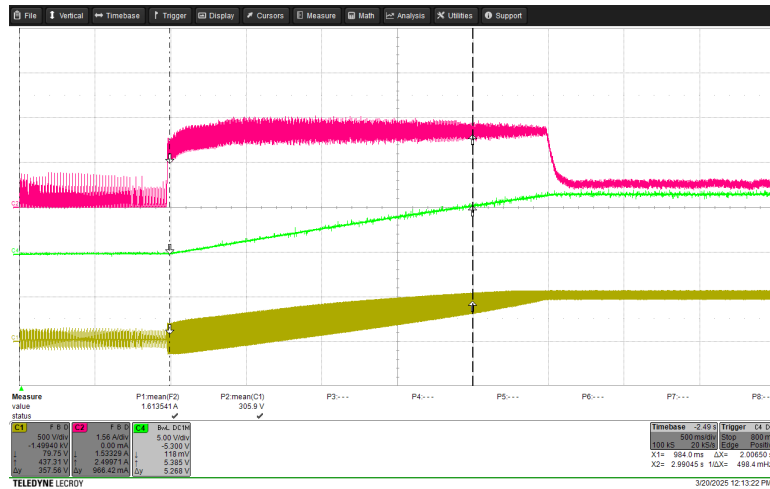


Figure 3: Image de l'oscilloscope pour mesurer T_m

3.3 Dimensionnement

On peut désormais déterminer les gains du régulateur. Comme la MCC est approximée comme un intégrateur, on a que $T_n = 4 \cdot T_{pE} = 0.1239$. A l'aide de l'optimum symétrique, on obtient $T_i = 8 \cdot \frac{T_{pE}^2}{T_m} = 0.0041$.

On obtient ainsi les gains du régulateur:

$$K_i = \frac{T_E}{T_i} = 0.2422$$

$$K_p = \frac{T_n - \frac{T_E}{2}}{T_i} = 29.8955$$

4 Simulations

On peut constater que le mode théorique (figure 4a) présente des dépassements de consigne ainsi qu'un temps de stabilisation plutôt long. L'introduction du correcteur de consigne (figure 4b) permet de mitiger ce dépassement mais augmente le temps mis à atteindre la consigne.

En activant le mode Standard (figure 5a) on constate que la limitation de la consigne n'a presque aucun effet, car elle n'agit uniquement si la consigne dépasse 2 p.u.. Le mode ARW augmente considérablement le temps de stabilisation, mais on remarque que la consigne n'est presque pas dépassée, et les termes proportionnel et intégrale du régulateur présentent moins de pics (plus de stabilité). Enfin le mode global semble très similaire au mode ARW, car la consigne ne dépasse ici pas 2 p.u..

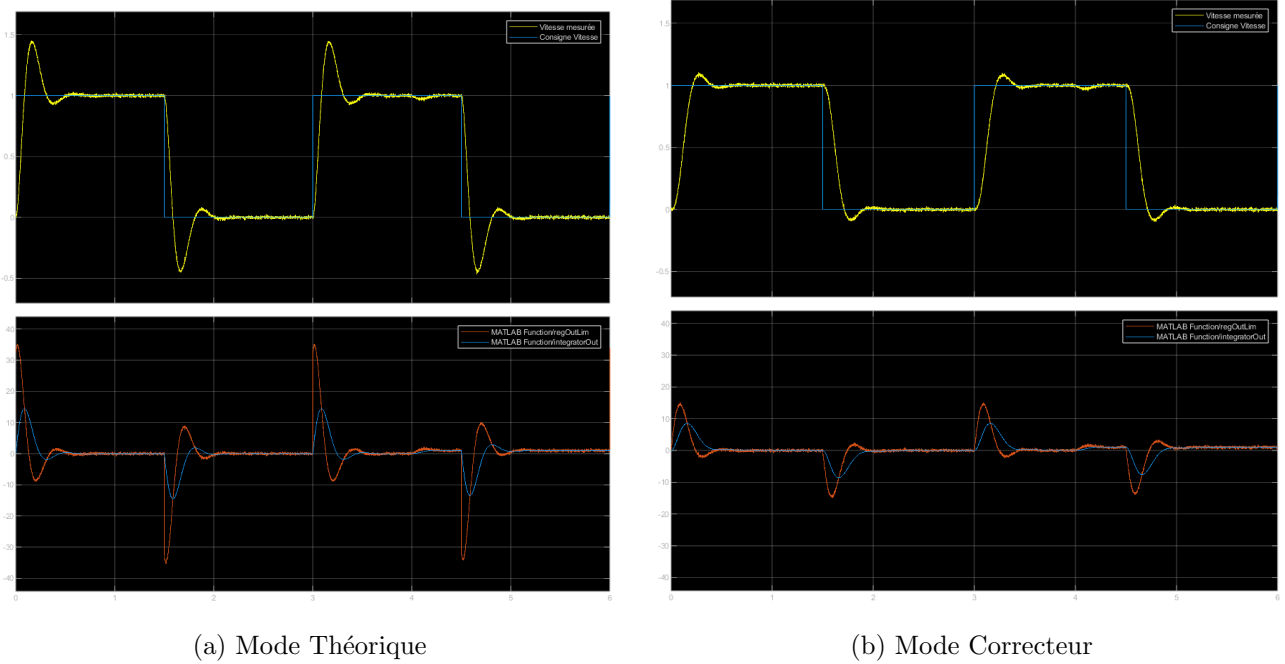


Figure 4: Simulations - Sans limitations

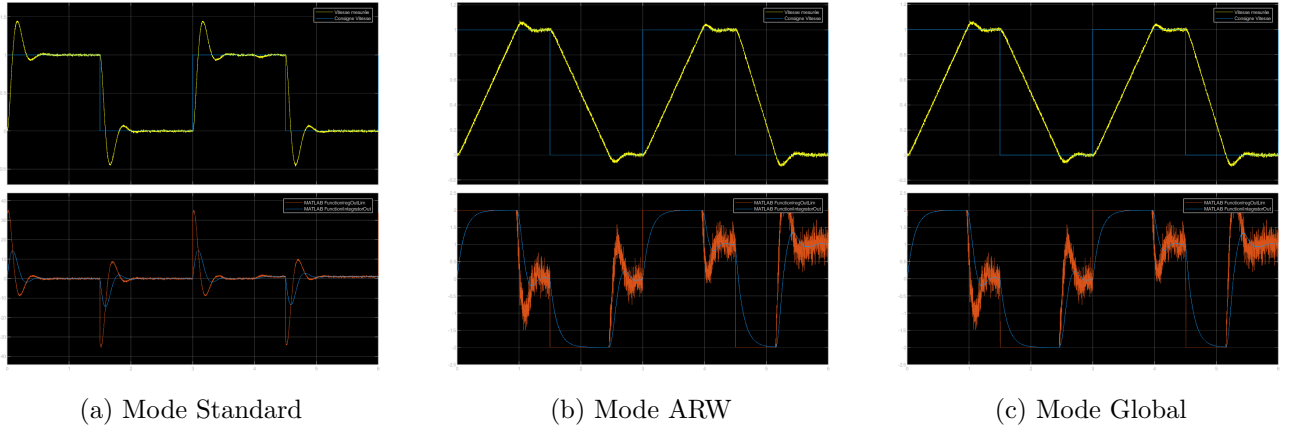
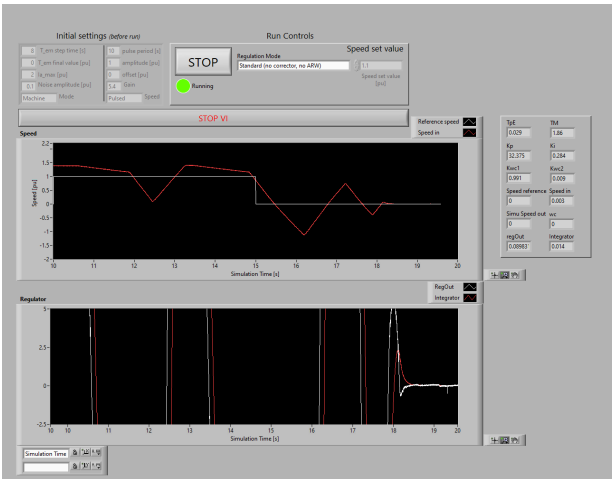


Figure 5: Simulations - Avec limitations

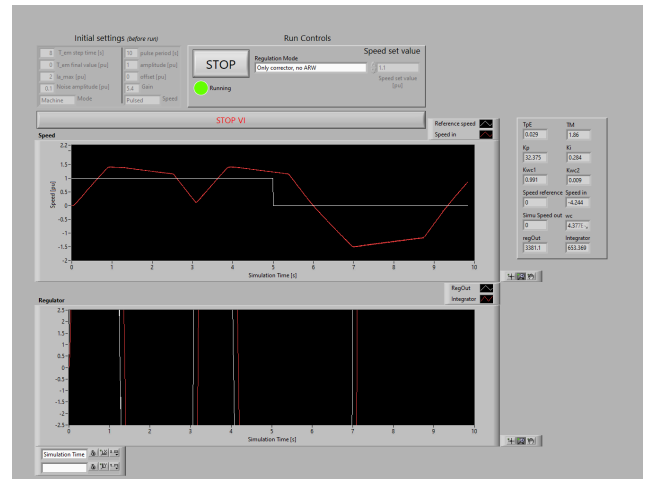
5 Mesures sur installation

Comme constaté dans la partie précédente, on observe que le mode standard (figure 6a) n'est pas satisfaisant en termes de performance, en effet le système oscille longtemps avant de se stabiliser lors d'un saut de consigne. En activant le mode correcteur (figure 6b), on remarque que l'oscillation est toujours présente, et le système est même instable. Ceci est dû au fait que les termes internes du régulateur ne sont pas limités, et la saturation de la consigne à 2 p.u. n'est pas suffisante pour mitiger cet effet.

Lorsque le mode ARW est utilisé (figure 7a), les termes internes (principalement intégral) sont limités et le système suit la référence correctement dans un temps raisonnable. L'ajout du correcteur de consigne dans le mode global (figure 7b) permet de réduire le dépassement.

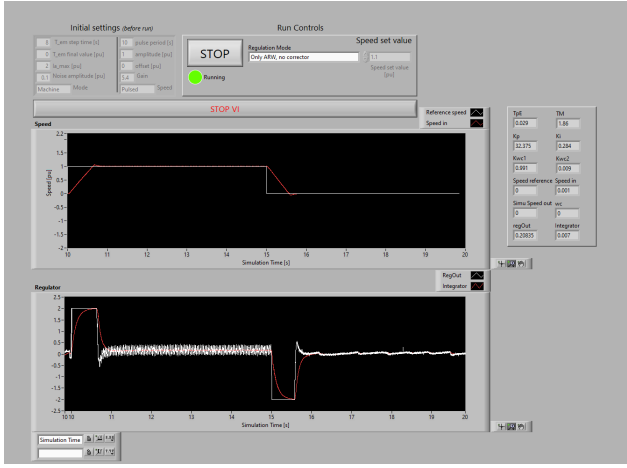


(a) Mode Standard

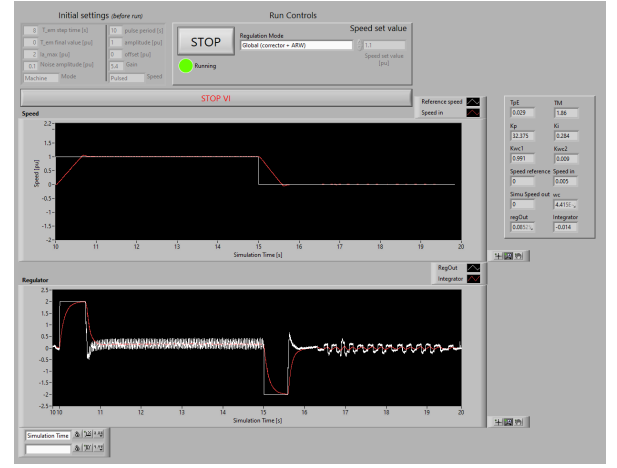


(b) Mode Correcteur

Figure 6: Vitesse du moteur, Sortie du contrôleur et Terme Intégrateur du contrôleur pour les modes **Standard** et **Correcteur**



(a) Mode Anti-Reset Windup



(b) Mode Global

Figure 7: Vitesse du moteur, Sortie du contrôleur et Terme Intégrateur du contrôleur pour les modes **Anti-Reset Windup** et **Global**

6 Conclusion

Nous avons développé avec succès un contrôle en vitesse pour une MCC, en caractérisant d'abord le système MCC+commande en courant, puis en calculant les gains d'un régulateur PI de commande en vitesse à l'aide du critère d'optimum symétrique, et enfin nous avons validé le régulateur sur le système complet. Ceci nous a permis de comprendre l'impact de différentes méthodes de régulation (ARW, compensation de consigne).