

## Projet : Contrôle commande d'un entraînement électrique

### Introduction

#### Objet

Il s'agit de montrer et de concevoir sous Matlab ou sur Scilab le fonctionnement d'un entraînement électrique, formé d'une machine électrique, d'un onduleur et de son contrôle commande.

Dans certaines parties du cours, on suppose que la machine a déjà atteint un régime établi, tandis que d'autres parties envisagent des régimes dynamiques quelconques.

Il s'agit ici de considérer les régimes dynamiques, dans le but de parvenir à guider peu à peu le système vers des régimes plus établis.

Les différents composants seront introduits progressivement dans le code.

Celui-ci sera testé à chaque étape et ses performances analysées.

Les dispositifs introduits à chaque étape seront :

1. Machine synchrone alimentée en pleine onde
2. Modulation de largeur d'impulsion
3. Contrôle commande en tension progressive
4. Contrôle commande en flux

On représentera les composants de puissance tels qu'ils sont définis.

Le système de contrôle commande sera conçu selon des critères de performances.

N.B. : L'énoncé peut comporter des indications non requises pour mener à bien le projet.

On notera qu'il fait partie du travail d'un ingénieur de discerner quelles sont les données pertinentes parmi celles disponibles ou à rechercher.

#### Dispositif

La machine électrique est triphasée synchrone à aimants et comporte 4 paires de pôles.

Elle entraîne une inertie.

Elle est alimentée par un onduleur triphasé à transistors et diodes de roue libre.

Les six transistors de l'onduleur sont commandés par une carte de commande comportant tous les moyens de calculs requis en temps réel.

Cette carte de contrôle reçoit à intervalle régulier les mesures de la tension d'alimentation continue de l'onduleur, des courants circulant dans les trois phases de la machine, et de l'angle dont le rotor a tourné.

#### Caractéristiques

Nombre de paires de pôles	$P_p = 4$
Résistance stator vue en Park	$R_s = 0,18 \, \Omega$
Inductance stator de Park directe	$L_d = 1,15 \, \text{mH}$
Inductance stator de Park quadratique	$L_q = 3,31 \, \text{mH}$

Induction rémanente des aimants	1,3 Tesla
Flux des aimants vu par l'ensemble des spires d'une phase	200 mWb
Couple maximum	$RC1 = 50 \text{ N.m}$
Vitesse mécanique de base	$W_{\text{base}} = 350 \text{ rad/s}$
Tension continue maximale de l'onduleur	$2.E_{\text{max}} = 800 \text{ Volts}$
Courant crête maximal dans les phases	55 A
Inertie entraînée	$J = 800.10^{-6}$

Toutes ces caractéristiques sont celles d'une machine réelle, à l'exception de l'inertie  $J$  qui a été fortement réduite pour permettre de voir évoluer rapidement la vitesse, tout en conservant des durées de simulation non contraignantes pour ce projet.

### Algorithme numérique temporel

Le calcul étant numérique, il faut choisir un pas temporel  $t_0$  pour définir les instants successifs auxquels on mènera les calculs. Ce pas doit être :

- Très petit devant les plus petits temps caractéristiques du système
- Pas trop petit pour que la simulation ne dure pas trop longtemps.

Les calculs se répétant beaucoup d'un instant de calcul sur le suivant, on choisit généralement de faire une boucle, chaque itération successive de cette boucle consistant à calculer les variables du système à un instant physique déterminé.

### Variables d'état

L'état d'un système à un instant donné dépend :

- Des entrées appliquées à ce système à cet instant là
- De l'état antérieur du système.

Sont particulièrement importantes à cet égard les « variables d'état », c'est-à-dire celles qui représentent des énergies présentes en un lieu, et qui ne peuvent donc pas varier brutalement.

A chaque simulation, on fera partir le système d'un état simple :

Source de tension, onduleur et machine électrique sont connectés entre eux.

Tous les transistors sont bloqués depuis longtemps ; le rotor est arrêté.

On rappelle que les aimants ont été magnétisés

avant même d'être montés dans le rotor, lors de la fabrication de la machine.

### Cœur du programme : calculs aux instants successifs

C'est le cœur du programme que l'on rédige normalement en premier, même s'il sera précédé de diverses initialisations dans le code exécutable.

Chaque pas de calcul (i) comporte les grandes étapes suivantes :

- Calcul des entrées du système au pas (i)
- A partir de ces entrées et des valeurs des variables d'état  $X(i)$  à cet instant (i) (Nous verrons tout à l'heure comment on dispose de ces variables d'état  $X(i)$ ), calcul des différentes variables au pas (i), selon l'ordre indiqué sur les schémas fonctionnels, jusqu'à arriver à calculer les dérivées des variables d'état  $[dX(i)/dt]$  au pas (i).
- Eventuellement calcul d'autres variables (i) dont on souhaite tracer les courbes.
- Calcul de chacune des variables d'état au pas (i+1), par intégration, par exemple selon la formule simple :  $X(i+1) = X(i) + [dX(i)/dt] \times [\text{pas temporel}]$ .

**C'est bien à la fin de chaque itération de la boucle que l'on procédera à toutes ces intégrations.**

Ainsi se termine le pas de calcul (i).

Quand on aborde le pas suivant, ce qui était  $X(i+1)$  au pas précédent devient  $X(i)$ .

C'est ainsi que l'on a cette connaissance annoncée de  $X(i)$ .

Pour le premier pas, on a initialisé  $X(1)$  avant d'entrer dans la boucle, qui commence donc à  $i=2$ .

**En veillant à la structure rigoureuse de la programmation,  
on évitera les bugs, et on gagnera ainsi beaucoup de temps.  
Par exemple, on évitera d'utiliser des grandeurs  
dont on n'a pas encore calculé la valeur.**

## Désignation des grandeurs

Au fur et à mesure, on peut placer les valeurs de chaque variable aux instants successifs dans un vecteur Matlab, particulièrement les variables d'état, et les variables dont on souhaite tracer les courbes.

Comme toujours en informatique,

on conseille de paramétrer au maximum les informations, c'est-à-dire de les représenter par des lettres dont on définit par ailleurs les valeurs numériques.

- Chaque information étant en général utilisée plusieurs fois, cela évite les oublis lorsque l'on veut en changer la valeur.
- Cela s'applique même aux constantes physiques du système, ne serait-ce que pour les modifier facilement dans le but de tester leur influence, lors de la mise au point du programme.

## Structure générale du programme

La structure générale du programme devient finalement :

1/ INITIALISATIONS :

- Entrée des valeurs des constantes physiques
- Entrée des constantes liées au pas temporel
- Eventuellement pseudo-déclarations de vecteurs Matlab pour en fixer le nombre d'éléments
- Initialisations des variables d'état au démarrage
- Eventuellement initialisations des fenêtres de courbes

2/ BOUCLE comprenant le cœur du programme tel que décrit précédemment

3/ TRACES DE COURBES

## Logiciel

Un programmera sous MATLAB simple. NE PAS utiliser Simulink.

## Compte-rendu

Le compte-rendu du projet comportera notamment :

- Les codes Matlab des programmes de simulation, dûment commentés.
- Les courbes obtenues en simulation, sauf indication contraire.
- Les réponses aux questions posées.

**Le compte-rendu sera mis sous la forme d'UN SEUL fichier .PDF,  
à l'exclusion de tout autre type de fichier.**

**Il sera envoyé exclusivement à l'adresse : [jacques.laeuffer@gmail.com](mailto:jacques.laeuffer@gmail.com)**

## **1. Machine synchrone**

### **Principe**

Dans un premier temps, on souhaite représenter sous Matlab cette machine électrique avec l'inertie, et l'alimenter par un onduleur triphasé dit « pleine onde ».

Chaque demi-pont délivre une tension carrée périodique, alternativement positive et négative, les trois tensions étant déphasées les unes des autres dans le temps de 120 degrés. Une façon simple de les programmer sous Matlab est par exemple :

$$\begin{aligned}Vo1(i) &= E * \text{sign}(\sin(W_v * t(i) + \pi)) \\ Vo2(i) &= E * \text{sign}(\sin(W_v * t(i) + \pi - 2 * \pi / 3)) \\ Vo3(i) &= E * \text{sign}(\sin(W_v * t(i) + \pi - 4 * \pi / 3))\end{aligned}$$

La pulsation des tensions sera de  $W_v = 70 \text{ rad/s}$ . Comme la machine est de type synchrone, le but sera que le rotor arrive à suivre. On verra si l'on y parvient.

Pour éviter les surintensités, la tension continue d'alimentation est réduite à  $2.E = 24 \text{ V}$ .

On mènera la simulation durant une centaine de millisecondes.

Pour la partie puissance du système :

- Le cœur de la puissance sera représenté sous forme Park.
- L'onduleur commandera les tensions triphasées, puisque ce sont celles-là qu'il peut effectivement commander.
- La machine sera chargée par un couple de frottement visqueux  $C_r = k \cdot W_e$ , avec  $W_e$  vitesse de rotation électrique en rad/s et  $k = 0,30$  coefficient de frottement visqueux.
- Dans les calculs de  $V_d$  et  $V_q$  par  $\text{Park}^{-1}$ , on pourra remplacer  $V_1, V_2, V_3$  par  $Vo1, Vo2, Vo3$ , car les termes en  $V_n$  s'annulent sous l'effet des trois angles.

Rédiger sous Matlab les équations correspondantes, en partant des sorties des intégrateurs comme indiqué dans « Cœur du programme : calculs aux instants successifs ».

Rédiger sous Matlab les initialisations.

### **Courbes**

On souhaite tracer en fonction du temps les variables suivantes :

Sur un 1<sup>er</sup> diagramme : Les tensions triphasées de l'onduleur  $Vo1, Vo2, Vo3$ . Pour mieux les distinguer sur le diagramme, on ajoutera +5 volts à la première, rien à la seconde, -5 volts à la dernière.

Sur un 2<sup>ème</sup> diagramme : Les tensions direct et quadratique de la machine  $V_d$  et  $V_q$ .

Sur un 3<sup>ème</sup> diagramme : Les flux direct et quadratique de la machine  $\Phi_{id}$  et  $\Phi_{iq}$ .

Sur un 4<sup>ème</sup> diagramme : Les courants direct et quadratique de la machine  $I_d$  et  $I_q$ .

Sur un 5<sup>ème</sup> diagramme : Les courants triphasés dans la machine  $I_1, I_2, I_3$ .


Sur un 6<sup>ème</sup> diagramme : Les grandeurs mécaniques, avec des échelles relatives appropriées : le couple  $C$ , la vitesse « électrique »  $W_e$ , l'angle « électrique »  $A$  du rotor.

Rédiger sous Matlab les instructions pour tracer ces courbes.

## Simulation

Lancer une 1<sup>ère</sup> simulation. Mettre au point.

## Débogage

On conseille de tracer d'entrée de jeu toutes les courbes proposées, car elles sont un excellent moyen de débogage. 

On pourra bien sûr aussi tracer d'autres courbes dans ce but.

Le débogage se fait en partant des entrées et en allant progressivement vers les sorties.

Chaque étape est un diagramme, sur lequel on peut vérifier la cohérence des données par rapport au diagramme précédent.

Cette vérification peut se faire en choisissant arbitrairement un instant particulier, et en refaisant - à la calculatrice - les calculs pour cet instant-là.

Une difficulté réside dans le fait que le système est rebouclé. Par exemple :

- l'angle A intervient dans les calculs de transformée de Park,
- We, Id, Iq interviennent dans les calculs des dérivées des flux.
- Dans un premier temps, en reprenant le schéma fonctionnel, on peut se contenter de vérifier la cohérence entre l'entrée et la sortie de chaque « boîte » de ce schéma.
- Il ne s'agit pas encore de valider l'exactitude des résultats, mais seulement l'exactitude de la partie de programme de calcul de cette étape.
- On peut le faire en choisissant un instant un peu arbitraire, on relève sur les courbes les valeurs des paramètres à cet instant, et on refait les calculs à la calculatrice pour vérifier leur cohérence.

Une méthode plus radicale consiste – dans un premier temps - à commencer par supposer que le résultat voulu est déjà obtenu :

- Entrer dans le calcul de Park non pas l'angle issu des intégrations, mais un angle tournant de façon synchrone avec les tensions appliquées.
- Ensuite bien sûr, il faut remplacer cet angle « a priori » par l'angle issu des intégrations pour obtenir le vrai résultat.

## Analyse

On notera le courant crête dans les phases,  
la vitesse moyenne obtenue,  
le couple moteur moyen,  
l'angle électrique A dont la machine a tourné en 20 ms.  
(Comme on s'intéresse à des ordres de grandeur, 2 chiffres significatifs suffisent.)  
On pourra rassembler ces valeurs dans un tableau  
que l'on complètera au fil des questions successives.

Analyser et commenter abondamment - par exemple sur une page - les courbes obtenues ;  
en exprimant en quoi on a atteint ou non les objectifs,  
en essayant de comprendre les causes physiques de ce que l'on voit.

N.B. : Il est inutile de simplement recopier l'énoncé du projet.

## Sans couple résistant

Supprimer le couple résistant et lancer une 2<sup>ème</sup> simulation.

Qu'observez-vous de nouveau ? A quoi cela est-il dû ?

(On ne reproduira PAS dans le compte-rendu les courbes de cette 2<sup>ème</sup> simulation.)

## **2. Modulation de largeur d'impulsion**

### **Principe**

Conserver le programme de la partie précédente sous un nom spécifique, et le recopier sous un nouveau nom comme base de départ pour cette nouvelle partie.

On souhaite maintenant reprendre l'ensemble du dispositif précédent, à ceci près que ses entrées Vo1, Vo2, Vo3 ne seront plus les entrées du système, mais que Vo1, Vo2, Vo3 résulteront des actions d'un onduleur avec modulation de largeur d'impulsion, afin de rendre les tensions de la machine plus sinusoïdales au cours du temps, et ainsi éviter les à-coups de couple.

Les nouvelles entrées du système sont les consignes de tension sinusoïdales à appliquer aux demi-ponts de l'onduleur. Elles peuvent s'écrire :

$$RVo1(i) = E \cdot \sin(W_v \cdot t(i) + \pi)$$

$$RVo2(i) = E \cdot \sin(W_v \cdot t(i) + \pi - 2 \cdot \pi/3)$$

$$RVo3(i) = E \cdot \sin(W_v \cdot t(i) + \pi - 4 \cdot \pi/3)$$

La pulsation des tensions sera de 220 rad/s.

Pour éviter les surintensités, la tension continue d'alimentation est limitée à  $2 \cdot E = 100 \text{ V}$ .  
On mènera la simulation durant vingt millisecondes.

On appliquera un couple résistant de frottement visqueux, mais avec  $k = 0,20$

### **Délais de commutation**

On choisit comme fréquence de découpage de l'onduleur 10 KHz,  
d'où une période de découpage  $t_p = 100 \mu\text{s}$ .

Puis l'on en déduit les délais à respecter entre le début de la période de découpage et les commutations d'un demi-pont.

On choisit que toutes les tensions soient à  $-E$  au début et à la fin de  $t_p$ .  
 $tr1$  est l'instant où la tension Vo1 monte, et  $tf1$  l'instant où elle descend.

On rappelle que :  $tr1 = t_p \cdot (1 - RVo1/E)/4$   
 $tf1 = t_p \cdot (3 + RVo1/E)/4$ . De même pour les autres demi-ponts.

### **Réalisation des calculs**

En général, dans les systèmes industriels, le microcontrôleur ou DSP dispose de toute une période de découpage pour mener à bien un seul calcul de régulation. Ainsi, si un échantillonnage des mesures des grandeurs de puissance est réalisé au début de la période (T), le calcul est mené durant cette période (T), et ces délais de commutation sont appliqués aux transistors durant la période (T+1).

N.B. : les échantillonnages-blocages des mesures seront réalisés dans la partie 3.

### **Algorithme informatique temporel avec calculs de contrôle**

En plus de la puissance, il s'agit maintenant de représenter aussi le contrôle commande de l'entraînement électrique.

Un pas de calcul de calcul plus fin que dans la partie précédente sera peut-être nécessaire, pour faire apparaître les commutations bien verticales sur les diagrammes.

#### **Tout ce que nous avons dit en introduction**

**concernant l'algorithme informatique temporel reste valable  
et doit continuer à être respecté,  
en particulier l'ordre des calculs dans la boucle.**

**Il s'agit d'un seul calcul temporel : ainsi, chacune des étapes déjà décrites pourra  
comporter à la fois des aspects puissance et des aspects contrôle commande.**

Dans cette partie et les suivantes, les calculs et synchronisations de contrôle commande qui vont être faits sous Matlab sont sensiblement les mêmes que ceux effectués en temps réel dans les microcontrôleurs, ou DSP et FPGA.

Les microcontrôleurs réalisant en général leur cycle de calcul une fois par période de découpage de l'onduleur, on va faire de même sous Matlab.

Cela signifie que, contrairement aux calculs de puissance qui évoluent à chaque pas temporel, les calculs de contrôle commande ne doivent être renouvelés qu'une seule fois par période de découpage.

Par ailleurs, contrairement au cas industriel, on n'a pas de contrainte temps réel sous Matlab : on peut faire l'ensemble des calculs de régulation sur un seul pas de calcul, lors d'un pas arbitrairement choisi, par exemple au milieu de la période de découpage.

Pour distinguer les pas où l'on va faire les calculs de régulation parmi les autres pas, on peut par exemple utiliser une instruction « if », avec une condition temporelle.

Pour simplifier la recopie des variables de contrôle lorsqu'elles n'ont pas à changer, on peut faire *précéder* ces calculs de régulation, à *tous* les pas, d'une recopie de ces variables de contrôle. Si, dans le déroulement ultérieur du pas de calcul, la condition « if » est réalisée, la recopie précédente est alors être écrasée.

## Séquençement des commutations

On peut, en plus du temps courant  $t(i)$ , définir un temps à l'intérieur de la période  $t_{int}(i)$ .

On commence donc la période avec  $V_{o1} = V_{o2} = V_{o3} = -E$ .

Au début de la période, on calcule les nouveaux délais qui seront à appliquer au cours de cette période, sur la base des calculs qui ont été menés au milieu de la période précédente.

A chaque pas de calcul, pour  $V_{o1}$ , on peut tester si  $t_{int}(i)$  est compris entre le moment où  $V_{o1}$  doit monter et le moment où  $V_{o1}$  doit redescendre, et attribuer  $+E$  ou  $-E$  à  $V_{o1}$  en conséquence. On fait de même pour  $V_{o2}$  et  $V_{o3}$ .

Attention, si l'on compare des temps qui sont représentés par des nombres réels, la comparaison entre des temps physiquement égaux peut basculer aléatoirement d'un côté ou de l'autre en fonction des arrondis. Dans ce cas, une solution consiste à ajouter - dans l'équation de comparaison - la moitié d'un pas de temps de calcul.

## Courbes

On donnera dans le compte-rendu

les diagrammes des grandeurs physiques données pour la question précédente, actualisés dans les nouvelles conditions, et avec les modifications suivantes :

- Sur le 1<sup>er</sup> diagramme : ajouter  $t_{int}$ ,  $tr1$  et  $tf1$ , et dilater la courbe pour ne représenter environ que 2. $t_p$ .
- Sur le 2<sup>ème</sup> diagramme : dilater la courbe pour ne représenter environ que 2. $t_p$ .
- Sur un 7<sup>ème</sup> diagramme : tracer  $RV_{o1}$ ,  $RV_{o2}$ ,  $RV_{o3}$ .

Rédiger sous Matlab les instructions pour tracer ces courbes.

## Simulation et analyse

Lancer la simulation. Mettre au point.

On notera le courant crête dans les phases, la vitesse établie, le couple moteur établi, l'angle électrique  $A$  dont la machine a tourné en 20 ms.

Analyser et commenter abondamment - par exemple sur une page - les courbes obtenues ; en exprimant en quoi on a atteint ou non les objectifs, en essayant de comprendre les causes physiques de ce que l'on voit.

### **3. Commande en tension progressive, et surveillance**

#### **Principe**

Conserver le programme de la partie précédente sous un nom spécifique, et le recopier sous un nouveau nom comme base de départ pour cette nouvelle partie.

On souhaite maintenant reprendre l'ensemble du dispositif précédent, à ceci près que les entrées précédentes  $RVo1$ ,  $RVo2$ ,  $RVo3$  ne seront plus les entrées du système, mais que  $RVo1$ ,  $RVo2$ ,  $RVo3$  résulteront d'une commande du système en tension progressive, pour monter à des vitesses plus importantes sans décrochements ni surintensités. En effet, les tensions doivent progresser notamment avec les vitesses.

Le nouveau système comportera aussi - au niveau du contrôle commande - une surveillance de l'intensité des courants.

Les tensions seront progressives à la fois en amplitude et en fréquence.

Pour éviter que la machine ne décroche,

il faut assurer un angle correct entre les champs stator et rotor, ce qui requiert :

- de connaître la position du rotor
- d'appliquer au stator un courant quadratique pour produire du couple.

On restera en dessous de la vitesse de base.

La fréquence de découpage de l'onduleur restera à 10 KHz.

La tension continue d'alimentation est maintenant portée à  $2.E = 800 \text{ V}$ .

On mènera la simulation durant vingt millisecondes.

On appliquera un couple résistant de frottement visqueux, mais avec  $k = 0,10$

#### **Calcul des tensions et échantillonnages**

Pour cela, les variables d'entrée du système seront maintenant  $RVd$  et  $RVq$ .

A partir d'elles, le contrôle calculera  $RV1$ ,  $RV2$ ,  $RV3$  par la transformation de Park-1.

Park-1 permettra notamment une prise en compte l'angle du rotor.

Pour maintenir stables les flux  $\Phi_{id}$  et  $\Phi_{iq}$ , leurs dérivées devront rester proches de zéro.

$Vd$  et  $Vq$  devront donc croître au cours du temps pour équilibrer la croissance de la vitesse  $W_e$ .

$Vd$  et  $Vq$  reproduisant  $RVd$  et  $RVq$ , on fera croître les nouvelles entrées  $RVd$  et  $RVq$  linéairement en fonction du temps.

-  $RVd$  partira de zéro et descendra (avec une pente négative donc) de  $-5000 \text{ V/s}$

-  $RVq$  partira de zéro et montera avec une pente de  $+ 5000 \text{ V/s}$

Le calcul par Park-1 de  $RV1$ ,  $RV2$ ,  $RV3$  ayant lieu dans le contrôle, celui-ci doit disposer d'une mesure  $MA$  de l'angle  $A$ .

On reproduira le système industriel qui échantillonne les mesures au début de chaque période de découpage.

On peut calculer une consigne de tension à appliquer au neutre de la machine,  $RVn$ , d'une façon un peu différente de celle du cours, et plus aisée en temps réel.

On calcule d'abord le MAX de  $RV1$ ,  $RV2$ ,  $RV3$ , puis le MIN de  $RV1$ ,  $RV2$ ,  $RV3$ , et l'on choisit  $RVn$  comme l'opposé de la moyenne des deux.

On en déduit les consignes en entrée de l'onduleur :  $RVo1$ ,  $RVo2$ ,  $RVo3$ .

On échantillonnera et bloquera aussi les mesures  $MI1$ ,  $MI2$ ,  $MI3$  des courants  $I1$ ,  $I2$ ,  $I3$  pour en surveiller les amplitudes.



## Algorithme informatique temporel

Conformément à ce qui est fait industriellement, on échantillonne donc toutes les mesures au début de chaque nouvelle période de découpage. Afin d'en disposer lors des calculs ultérieurs, et pour pouvoir les représenter aisément sur des courbes temporelles sous forme de créneaux successifs, on peut, aux autres pas de calcul, recopier la valeur du pas précédent : c'est d'ailleurs ce que l'on appelle le blocage d'une valeur échantillonnée.

Pour distinguer les pas d'échantillonnage parmi les autres pas, on peut par exemple utiliser une instruction « if » avec une condition temporelle adaptée.

## Courbes

On donnera dans le compte-rendu les diagrammes des grandeurs physiques données pour la question précédente, actualisés dans les nouvelles conditions, et avec les modifications suivantes :

Sur le 2<sup>ème</sup> diagramme : supprimer  $V_d$  et  $V_q$   
ajouter  $RV_d$ ,  $RV_q$  et  $-RV_n$ ,  
représenter les 20 ms de la simulation.

Sur le 5<sup>ème</sup> diagramme : ajouter  $MI_1$ ,  $MI_2$ ,  $MI_3$

Sur un 8<sup>ème</sup> diagramme : tracer  $RV_1$ ,  $RV_2$ ,  $RV_3$  et  $-RV_n$

## Simulation et analyse

Lancer la simulation. Mettre au point.

On notera le courant crête dans les phases, la vitesse finale, le couple moteur maximal, l'angle électrique  $A$  dont la machine a tourné en 7 ms.

Analyser et commenter abondamment - par exemple sur une page - les courbes obtenues ;  
en exprimant en quoi on a atteint ou non les objectifs,  
en essayant de comprendre les causes physiques de ce que l'on voit.

## **4. Commande en flux**

### **Principe**

Conserver le programme de la partie précédente sous un nom spécifique, et le recopier sous un nouveau nom comme base de départ pour cette nouvelle partie.

On souhaite maintenant reprendre l'ensemble du dispositif précédent, à ceci près que les entrées précédentes  $V_d$ ,  $V_q$  ne seront plus les entrées du système, mais que  $V_d$ ,  $V_q$  résulteront d'une régulation des flux  $\Phi_{id}$  et  $\Phi_{iq}$  dans la machine, pour commander de façon précise et optimisée le couple  $C$

Ces régulations sont faites en boucle fermée, sur la base d'estimations des flux  $M\Phi_{id}$  et  $M\Phi_{iq}$ , issues des estimations  $MI_d$  et  $MI_q$ , elles-mêmes issues par Park des mesures  $MI_1$ ,  $MI_2$ ,  $MI_3$ .

Ce calcul Park utilise l'estimation de l'angle  $MA$  déjà élaborée.

$R\Phi_{id}$  et  $R\Phi_{iq}$  sont respectivement comparés à  $M\Phi_{id}$  et  $M\Phi_{iq}$ . Les erreurs sont amplifiées par des régulateurs pour élaborer les consignes  $RV_d$  et  $RV_q$ .

### **Gains croisés**

Dans la puissance, le flux  $\Phi_{id}$  est commandé par  $V_d$ , mais aussi par  $(W_e.\Phi_{iq})$ .

De même, le flux  $\Phi_{iq}$  est commandé par  $V_q$ , mais aussi par  $(-W_e.\Phi_{id})$ .

Pour améliorer les performances de la régulation, on compense ces termes dans  $RV_d$  et  $RV_q$  par respectivement  $(-MWe . M\Phi_{iq})$  et  $(+MWe . M\Phi_{id})$ .

$MWe$  est une estimation de  $W_e$  réalisée par dérivation de  $MA$  :

$$MWe(i) = [A(i) - A(i-1)] / t_p$$

### **Principe des régulateurs**

Les régulateurs ont pour rôle de minimiser les erreurs de boucle, mais tout en assurant la stabilité de ces boucles.

Des gains très faibles conduiraient à ce que  $RV_d$  et  $RV_q$  restent proches de zéro ; l'onduleur et la machine ne seraient pas commandés.

Des gains très élevés conduiraient à une action de très grande amplitude sur l'onduleur et la machine, avant même que l'on est le temps de la mesurer. A l'issue du délai de mesure, on s'apercevrait que l'on a dépassé les consignes, et l'on appliquerait alors de fortes consignes inverses, exagérées dans l'autre sens. On amorcerait ainsi une oscillation aussi violente qu'incontrôlée.

Pour régler les gains des régulateurs, il convient donc d'abord d'analyser les dynamiques de réponse du système déjà élaboré,

## Fonction de transfert sans retards purs, de Srd à MPhid et de Srq à MPhiq

En négligeant dans un premier temps les retards purs introduits par les échantillonnages, les fonctions de transfert déjà établies, pour l'axe d, s'analysent comme suit :

- De RVd à Vd : à peu près un gain unité.
- De Vd à DPhid : un gain unité, mais avec les perturbations (We.Phiq) et (-Rs.Id).
- De la sortie Srd du régulateur à RVd : un gain unité, mais avec une *compensation* de la perturbation (We.Phiq)
- DONC de Srd à DPhid : à peu près un gain unité.
- De DPhid à Phid : une intégration.
- De Phid à MPhid est à peu près un gain unité.
- DONC de Srd à MPhid : à peu près une intégration, sans aucun autre gain, c'est-à-dire un gain décroissant avec la fréquence, au premier ordre, et qui vaut un lorsque la pulsation vaut un radian par seconde.

Ainsi :  $(MPhid / Srd) = 1/p$  , avec  $p$  opérateur de Laplace

On rappelle qu'un diagramme de Bode en amplitude est une représentation d'une fonction de transfert, ayant en ordonnée l'amplitude de cette fonction de transfert, par exemple ici le module :  $|MPhid / Srd|$ , ayant en abscisse la pulsation  $p$  en radians par seconde, les deux étant en échelle logarithmique. Pour  $|MPhid / Srd|$ , il s'agit donc d'une droite de pente moins un, et passant par l'origine.

Il est va de façon similaire pour l'axe q.

## Prise en compte des retards purs dans les fonctions de transfert

Par ailleurs, les fonctions de transfert précédentes comportent des retards purs :

- L'évolution des courants n'est prise en compte dans les mesures qu'une seule fois par période  $t_p$ .
- Les calculs de régulation prennent une période  $t_p$  entière.
- Le système de modulation de largeur d'impulsion n'applique les nouvelles consignes que progressivement au cours de la période  $t_p$  suivante.

Il en résulte un retard pur d'environ  $(2.t_p)$ , dont la fréquence équivalente est

$$f(\text{retard pur}) = 1/(2 \cdot t_p) = 1/(2 \cdot 100 \text{ microsecondes}) = 5 \text{ KHz},$$
$$\text{de pulsation équivalente : } W(\text{retard pur}) = 2 \cdot \pi \cdot 5 \text{ KHz} = 31,4 \text{ Krad/s}.$$

La théorie des systèmes échantillonnés indique, pour éviter les effets de ces retards purs, que la fréquence de coupure  $f_c$  de la boucle ouverte

– c'est-à-dire la fréquence à laquelle le gain de la boucle ouverte tombe à un – doit être au moins un facteur 6 en-dessous de  $f(\text{retard pur})$

$$\text{soit : } f_c = 5 \text{ KHz} / 6 = 830 \text{ Hz}$$

$$\text{soit une pulsation de coupure : } W_c = 31,4 \text{ Krad/s} / 6 = 5,24 \text{ Krad/s}$$

On rappelle que la fonction de transfert en boucle ouverte est celle allant

de l'erreur de la boucle fermée jusqu'à la mesure qui est comparée à la consigne.

Ici, la fonction de transfert en boucle ouverte va donc

de EPhid à MPhid , ou bien de EPhiq à MPhiq.

## Calcul du gain proportionnel $G_p$

Dans un premier temps, on va calculer le gain proportionnel  $G_p$   
en maintenant le gain intégral  $G_i$  à zéro.

$$(S_{rd}/E_{Phid}) = G_p$$

$(M_{Phid}/S_{rd}) = 1/p$ , si les retards sont négligeables

Donc :  $(M_{Phid}/E_{Phid}) = G_p / p$ , fonction de transfert en boucle ouverte.

Pour obtenir les meilleures performances, on appliquera la bande passante la plus élevée possible, c'est-à-dire  $(M_{Phid}/E_{Phid}) = 1$  lorsque  $p = 5,24 \text{ Krad/s}$

Donc  $G_p = 5,24 \text{ Krad/s} \approx 5000 \text{ rad/s}$

## Schémas-blocs et diagramme de Bode

Dessiner des schémas-bloc successifs de la fonction de transfert de la boucle de flux direct : (pour gagner du temps, on peut si l'on veut les dessiner à la main, et en mettre une photo dans le compte-rendu .pdf.)

- 1/ en ne représentant les éléments périphériques que comme des perturbations
- 2/ en supprimant ces perturbations
- 3/ en simplifiant encore.

N.B. : Même si ces simplifications ont un caractère assez trivial, il est bon de les mener pas à pas pour s'assurer d'une bonne compréhension physique.

Dessiner le diagramme de Bode asymptotique en amplitude résultant de la fonction de transfert en boucle ouverte, en indiquant les différentes amplitudes et pulsations caractéristiques.

## Gain intégral $G_i$

En régime établi,  $Phid$  est constant, et par conséquent  $D_{Phid} = 0$ .

Avec seulement un gain proportionnel  $G_p$ ,

- les perturbations non compensées, comme  $(-R_s.Id)$ ,
  - ou imparfaitement compensées comme  $(M_{We}.M_{Phiq})$ ,
  - et le fait que le gain  $G_p$  n'est pas infini,
- conduisent à une erreur  $E_{Phid}$  non nulle, même en régime établi.  
Il en va de façon similaire pour l'axe  $q$ .

On dit qu'il y a une « erreur statique ».

Pour l'annuler, il convient qu'à fréquence nulle le régulateur ait un gain infini.

Mais cela ne doit pas déstabiliser la boucle aux plus hautes fréquences.

La solution consiste à ajouter un gain intégral dans le régulateur, constituant ainsi un régulateur « proportionnel-intégral ».

Pour éviter que ce gain intégral ne ramène du retard de phase à la fréquence de coupure de la boucle, on considère que la partie intégrale du gain du régulateur doit devenir inférieure à sa partie proportionnelle à une fréquence maximale égale à la fréquence de coupure de la boucle divisée par 5.

N.B. : Les intégrations dans les régulateurs n'étant faites qu'une fois par période  $t_p$ , c'est  $t_p$  qui intervient alors dans le calcul d'intégration.

Cependant, pour simplifier la programmation, on appliquera  $G_i = 0$

## Code

On se limitera au cas où l'on reste en-dessous de la vitesse de base.

Par conséquent, pour une consigne de couple RC, on appliquera :

$$R_{Phid} = \Phi_{im} \quad ; \quad R_{Phiq} = (L_q.RC) / (\Phi_p.\Phi_{im})$$

Compléter le code Matlab,

en respectant toujours la règle de partir de la sortie des intégrateurs,  
que ces intégrateurs se situent dans la partie puissance ou dans la partie contrôle.

La tension continue d'alimentation de l'onduleur est toujours  $2.E = 800$  Volts.

On n'appliquera PAS de couple résistant de frottement visqueux.

On souhaite mener la simulation sur 7 ms :

- De 0 à 500 microsecondes :  
n'appliquer aucune tension à la machine pour la laisser à l'arrêt.
- De 500 microsecondes à 1500 microsecondes :  
faire évoluer le couple continuellement de zéro jusqu'au couple maximum possible
- Appliquer ce couple maximum jusqu'à 7 ms.

Ecrire le code Matlab correspondant.

## Courbes

On donnera dans le compte-rendu les diagrammes des grandeurs physiques  
données pour la question précédente, actualisés dans les nouvelles conditions,  
et avec les modifications suivantes :

Sur le 3<sup>ème</sup> diagramme : ajouter  $R_{Phid}$  et  $R_{Phiq}$ ,  $M_{Phid}$  et  $M_{Phiq}$

Sur le 4<sup>ème</sup> diagramme : ajouter  $M_{Id}$  et  $M_{Iq}$

Sur le 6<sup>ème</sup> diagramme : ajouter  $M_A$ ,  $R_C$ ,  $M_{We}$ , la puissance électrique absorbée  $P$

## Simulation et analyse

Lancer la simulation. Mettre au point.

On notera le courant crête dans les phases, la vitesse finale, le couple moteur établi,  
l'angle électrique  $\theta$  dont la machine a tourné en 7 ms.

Analyser et commenter abondamment - par exemple sur une page - les courbes obtenues ;  
en exprimant en quoi on a atteint ou non les objectifs,  
en essayant de comprendre les causes physiques de ce que l'on voit.

## ANNEXE

