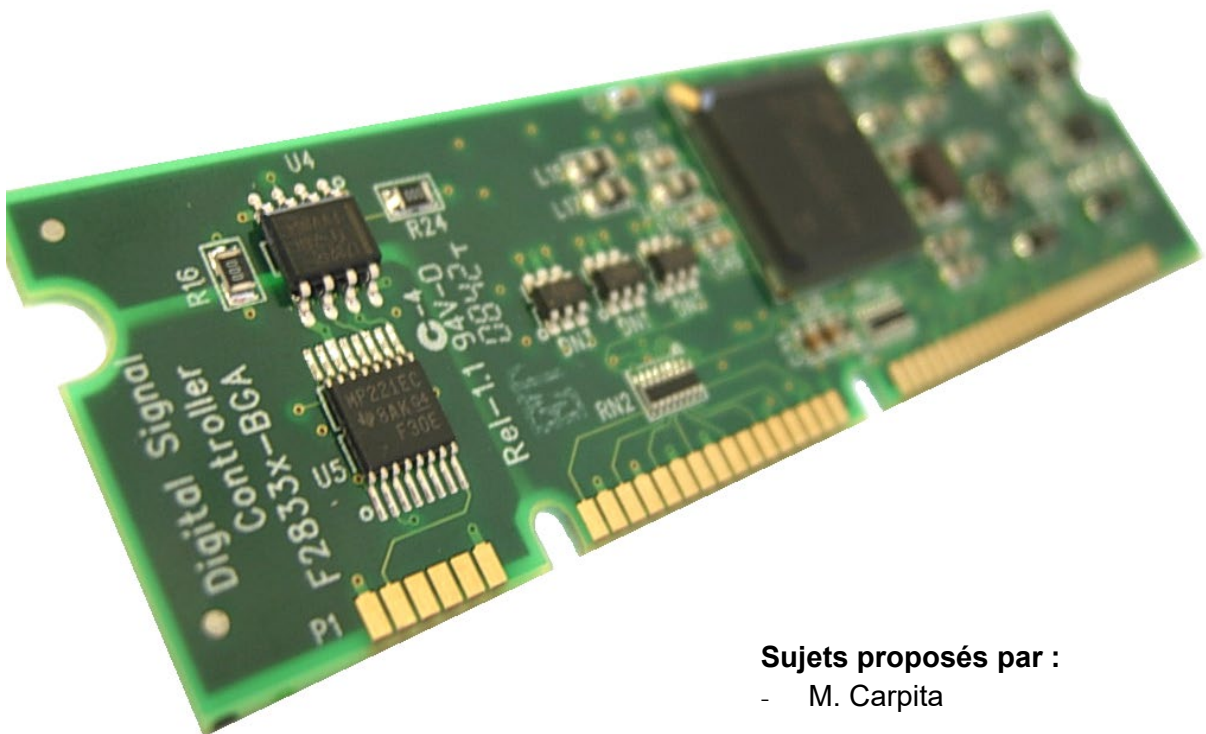


MA_PowELSYS

Synchronisation de l'onduleur triphasé avec le réseau électrique



Sujets proposés par :

- M. Carpita

Professeur :

- Mauro Carpita

DONNÉE SÉANCE 3 : SYNCHRONISATION DE L'ONDULEUR TRIPHASÉ AVEC LE RÉSEAU ÉLECTRIQUE**1. Introduction****1.1 Principe de la synchronisation et but du laboratoire**

L'idée du laboratoire est de synchroniser l'onduleur triphasé avec le réseau électrique. La technique de synchronisation utilisée est la SRF-PLL vue en cours dont le schéma est rappelé à la Figure 1.

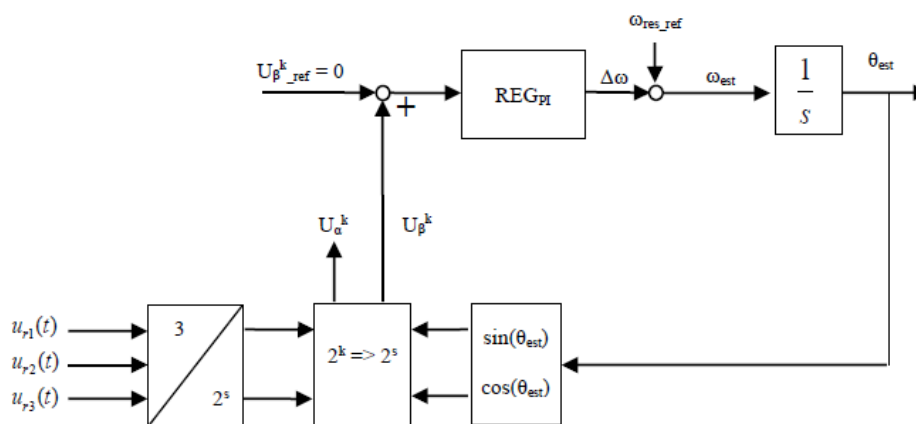


Figure 1. Schéma bloc du système de synchronisation SRF-PLL

Le but de cette séance est donc de programmer cette technique de synchronisation dans le DSP.

1.2 Programmation

Le but est de compléter le code manquant, à partir de celui de la seconde séance, afin de faire fonctionner le système.

Les fichiers à compléter :

- main.c
- Adc.c

Les instructions pour la modification du code sont à la suite de cette donnée.

1.3 Mesures

Les mesures sur l'installation didactique vont permettre de tester le code et de pouvoir le corriger si besoin.

| | | |
|------------------------------|-----------------|------------|
| Temps à disposition : | Programmation : | 1.5 heures |
| | Mesures : | 1.0 heures |

2. Programmation

Chargez le projet de la deuxième séance dans Code Composer. Faites ensuite une copie de celui-ci dans le « Project Explorer » afin de pouvoir changer son nom. Fermez la version de la séance 2 en faisant un clic droit puis « Close Project ».

Pour ce nouveau laboratoire, vous allez devoir utiliser un nouveau fichier « main.c ». Pour commencer, il faut supprimer le « main.c » actuel. Ce nouveau fichier comporte certains changements tels que des ajouts de *defines* sous *inputs/outputs*, de variables globales sous *Global vars* et modification de la partie *Commande Ucm*. Ce fichier se trouve dans l'onglet des fichiers Teams liés au laboratoire n° 3.

2.1 Modifier le fichier Main.c

ATTENTION, Certains points équivalents à ceux du labo 2 sont à réécrire dans ce nouveau fichier main.c.

Au début du fichier se trouve plusieurs « *define* » et variables globales utiles à l'écriture du code. Il n'y a donc pas obligation de les créer.

2.2 Complétez la routine d'interruption de l'ADC

2.2.1) Mesures U :

Pour la PLL, on veut mesurer la tension du réseau électrique. Ces mesures sont réalisées via la carte de mesure de la maquette.

Les résultats des mesures de tensions sont obtenus grâce au registre ADCRESULT de l'ADC (consultez le datasheet afin de comprendre son fonctionnement). La correspondance des registres avec les mesures de tension est indiquée ci-après :

Phase L1 de l'onduleur → ADCRESULT0

Phase L2 de l'onduleur → ADCRESULT1

Phase L3 de l'onduleur → ADCRESULT2

La commande pour récupérer la valeur du registre ADCRESULTx est :

« *AdcRegs.ADCRESULTx* »

ATTENTION : la mesure est donnée par un registre de 16 bits, dont certains bits sont réservés. Elle ne correspond pas directement à la valeur mesurée. Il faut donc décaler la valeur de ADCRESULTx de 4 bits vers la droite.

Modifiez également le registre ADCMAXCONV pour activer et indiquer le nombre de conversions de l'ADC, dans le fichier Adc.c.

3 conversion du coup 2 (ligne 86) validé par prof

2.2.2) Synchronisation PLL (procédure récapitulée sur le flow chart à la Figure 3) :

- a) A partir des mesures de tension du réseau, programmer la transformation de Clarke. Pour éviter des divisions dans le code, utilisez les *defines* de la section :

```
/*-----**
**                               inputs/outputs          **
**-----*/
```

- b) Dans la routine, pour ne pas devoir répéter plusieurs fois les fonctions « cos () » et « sin () », qui demandent beaucoup de temps d'exécution, on suggère de définir deux variables « cos_theta » et « sin_theta » à utiliser dans la suite du code.
- c) A présent que les variables « Ualpha », « Ubeta », « cos_theta » et « sin_theta » sont définies, effectuer la transformée de Park.
- d) Créer le régulateur PI et compenser la pulsation du réseau avec la valeur de sortie du régulateur. On suggère d'utiliser l'approche du « critère symétrique avec choix de la fréquence de coupure » (voir polycopiés du cours). La discrétisation du régulateur se fait à l'aide de la méthode de « Tustin ».
- [création de Ti et Tn, 3-49 pour le régulateur l'équation aux différences](#)
- e) Intégrer le signal pour obtenir theta (on suggère d'utiliser la méthode des trapèzes).
- [trapèze c'est Tustin](#)

Le régulateur PI est basé sur Tn et Ti, pour l'implémentation en C, c'est des coefficients KP et Ki

Cours 3, slide 3-45

$K_i = 1/T_i$

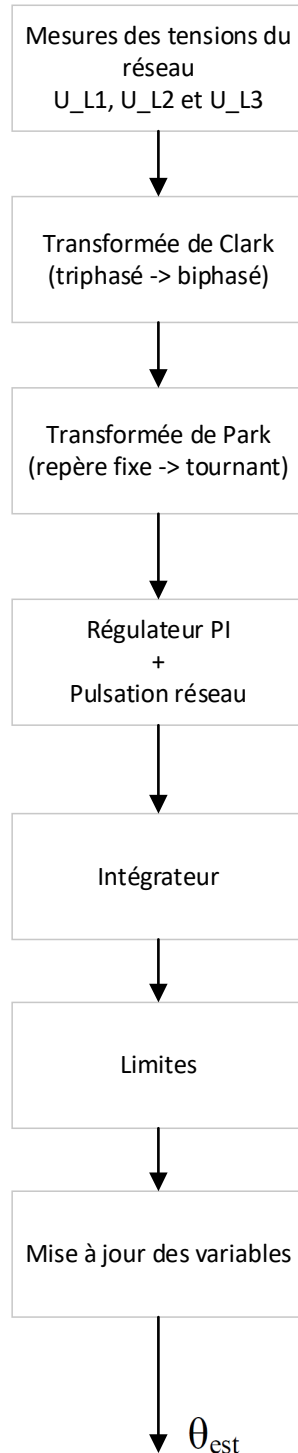
$K_p = T_n * K_i$

$T_i = T_{pe}^2 * U * \alpha^3$ (U phaseur spatial du réseau Ud)

$T_n = 8 * T_{pe}$

$T_{pe} = 1/f$

Figure 2 : Flow chart pour la Synchronisation de la SRF-PLL



3. Mesures

Les mesures vont consister à vérifier que vos tensions de sortie de votre onduleur sont synchronisées avec les tensions du réseau. Vous devrez faire vos mesures via les sondes différentielles de tension et l'oscilloscope (éventuellement les sondes de courants).

Faites un schéma de votre câblage. Le montrer au responsable de laboratoire avant de commencer à câbler.

ATTENTION, on mesure une tension dangereuse !!!

Il ne faut donc pas toucher les parties conductrices de la carte de mesure lors du branchement des fiches bananes reliées au réseau électrique !! Dans le doute, appeler le responsable de laboratoire.



Après avoir lu le message en rouge ci-dessus, faites le montage pour les mesures.

Expliquez dans le rapport si votre maquette fait le travail demandé.