

# Filière Systèmes industriels


Orientation Power & Control


## Travail de bachelor Diplôme 2022

*Daniel Héritier*

*Onduleur PV à base de ANPC*

 *Professeur*  
Stephan Kenzelmann


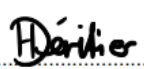
 *Expert*  
Rinaldo Arnold

 *Date de la remise du rapport*  
19.08.2022



Filière / Studiengang <b>SYND</b>	Année académique / Studienjahr <b>2021-22</b>	No TB / Nr. BA <b>PC/2022/90</b>
Mandant / Auftraggeber <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Etablissement partenaire Partnerinstitution	Etudiant / Student <b>Daniel Héritier</b> Professeur / Dozent <b>Stephan Kenzelmann</b>	Lieu d'exécution / Ausführungsort <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Etablissement partenaire Partnerinstitution
Travail confidentiel / vertrauliche Arbeit <input type="checkbox"/> oui / ja <input checked="" type="checkbox"/> non / nein	Expert / Experte (données complètes) <b>Rinaldo Arnold</b> - rinaldo.arnold@ch.abb.com ABB Schweiz AG, Austrasse, 5300 Turgi	

Titre / Titel  <b>Onduleur PV à base de ANPC</b>
Description / Beschreibung  Pour l'environnement POETIC, plateforme de contrôle et "fast prototyping", nous avons développé des modules de puissance type ANPC, un convertisseur multiniveau à des propriétés de redondance intéressant. Le travail de l'étudiant consiste dans l'élaboration des méthodes de modulation et des algorithmes de contrôle pour l'injection de courant dans le réseau triphasé. Après simulation et l'implémentation sur la plateforme POETIC, la plateforme Hardware-in-the-loop de Typhoon permettra de valider l'algorithme. Par la suite, la validation sur le prototype avec cartes de puissance ANPC aura lieu, avec tous les aspects de mesure de pertes et performance.  Objectifs / Ziele — Recherche littérature (modulation de l'ANPC, contrôle 3-niveau, injection réseaux) — Simulation Plecs (Source DC + ANPC + Injection Réseau). Simulations de défauts — Détermination de pertes et limitations en fonction des algorithmes et des configurations (Si, SiC, Hybride) — Détermination des procédures d'arrêt sur en cas de défaillance. — Simulation Typhoon HIL (Source DC + ANPC + Injection Réseau) — Programmation POETIC et méthodes de modulation — Simulation Typhoon + POETIC — Expérimentation avec réseau industriel, mesure et validation. — Documentation.

Signature ou visa / Unterschrift oder Visum  Responsable de l'orientation / Leiter der Vertiefungsrichtung:    <sup>1</sup> Etudiant / Student :  	Délais / Termine  Attribution du thème / Ausgabe des Auftrags: <b>16.05.2022</b>  Présentation intermédiaire / Zwischenpräsentation: <b>20-21.06.2022</b>  Remise du rapport final / Abgabe des Schlussberichts: <b>19.08.22, 12:00</b>  Expositions / Ausstellungen der Diplomarbeiten: <b>24-26.08.2022</b>  Défense orale / Mündliche Verfechtung: <b>Semaine/Woche 36 (5-9.09.2022)</b>
---	--

<sup>1</sup> Par sa signature, l'étudiant-e s'engage à respecter strictement la directive DI.1.2.02.07 liée au travail de diplôme.  
Durch seine Unterschrift verpflichtet sich der/die Student/in, sich an die Richtlinie DI.1.2.02.07 der Diplomarbeit zu halten.







## Travail de diplôme | édition 2022 |



Filière  
Système industriel (SYND)

Domaine d'application  
Power & Control

Professeur responsable  
Stephan Kenzelmann  
stephan.kenzelmann@hevs.ch

## Onduleur PV à base de ANPC



Diplômant Daniel Hérítier

### Objectif du projet

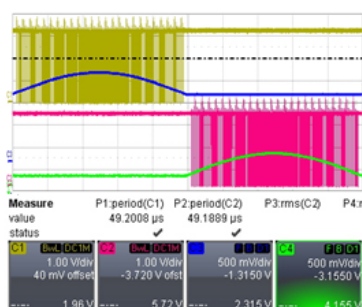
L'objectif du projet est de développer la partie software avec Poetic pour un onduleur triphasé ANPC qui peut interconnecter des panneaux PV avec le réseau. L'utilisateur choisit une référence de fonctionnement et le système doit automatiquement se régler sur cette consigne.

### Méthodes | Expériences | Résultats

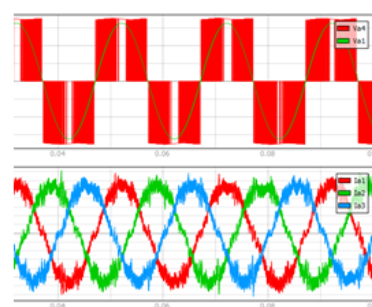
Pour atteindre les objectifs voulus, il faut commencer par faire des recherches sur le fonctionnement ANPC. Les avantages de cette topologie sont que lors d'un défaut il est possible de continuer à fonctionner à puissance réduite. Une fois le système bien compris on va pouvoir passer vers des simulations et évaluer les performances selon différents paramètres. Lorsque le fonctionnement correspond au résultat espéré, il sera possible de passer sur la programmation du système avec Poetic. Le programme est responsable pour la partie contrôle du système. Il gère les défauts et alarmes mais également la régulation pour atteindre les consignes de l'utilisateur. En sortie du régulateur on envoie les informations vers l'onduleur pour lui indiquer comment fonctionner.

En utilisant la méthode Hardware in the Loop avec Typhoon il sera possible de tester le code en détail et trouver les problèmes sans danger avant de brancher les composants réels. Avec Poetic et Typhoon il sera très pratique pour tenter des améliorations et de les tester rapidement.

Une fois que ces simulations sont terminées il sera possible de passer sur l'installation réel et de tester le fonctionnement en augmentant progressivement les puissances transférées.



Signaux 'codés' sur les deux fibres optiques A et B avec leurs signaux de modulations respectifs. Ceci permet d'envoyer les informations à l'onduleur.



Le 1er graphique est la tension de sortie sur la 1ère phase de l'onduleur avec la tension de réseau.  
2ème graphique sont les trois courants de sorties.



# Table des matières

1	Remerciements .....	1
2	Introduction .....	1
2.1	Bases du système.....	3
3	But et Objectif.....	5
3.1	Les étapes du projet.....	6
4	Recherche littérature .....	6
5	State machine des cartes de puissance.....	7
5.1	Explications State Machine des cartes de puissance.....	8
5.2	Explication du signal Desat .....	9
5.3	Démarche en cas de défaut .....	9
6	Transmission des paramètres de modulation vers l'onduleur .....	13
6.1	Modulateur .....	13
6.2	Signaux pour les modulateurs PWMA et PWMB.....	14
6.3	Transmission des données de modulation vers l'onduleur.....	15
6.4	Transmission des données, Cas Concret .....	16
6.5	Vue en détail des transmissions des données .....	18
7	Contrôle du système .....	18
7.1	PWM .....	18
7.1.1	Calcul pour la régulation .....	19
7.2	Space Vector Modulation .....	20
7.2.1	Fonctionnement.....	20
7.2.2	Régulation pour SVPWM.....	21
7.2.3	Calcul SVPWM .....	21
7.2.4	Comment choisir les vecteurs à utiliser pour reproduire le $v_{ref}$ ? .....	22
7.3	Outputs avec SVPWM .....	24
7.3.1	Activation SVPWM_1, version simple.....	24
7.3.2	Activation SVPWM_2, version complexe :.....	24
7.3.3	Fonctionnement SVPWM_2 .....	25
7.4	Comparaison des méthodes SVPWM_1 et SVPWM_2 .....	26
7.5	Fichier Excel pour les tests SVPWM au départ.....	27
8	Simulation Plecs (Source DC + ANPC + Injection Réseau). .....	28
8.1	ANPC Monophasé .....	28
8.2	ANPC Triphasé .....	29
8.3	Régulation .....	29
8.4	Résultat protection de la state machine .....	32

9	Détermination de pertes et limitations en fonction des algorithmes et des configurations (Si, SiC, Hybride) .....	32
9.1	Résultats Pertes.....	33
9.1.1	Mosfet.....	33
9.1.2	IGBT .....	34
9.1.3	Hybride .....	34
9.2	Conclusion .....	35
10	Programmation POETIC CPU et méthodes de modulation.....	36
10.1	State Machine du Système .....	36
10.2	Programme CPU.....	37
10.2.1	Code important sur le CPU Poetic.....	38
10.2.2	Code Space Vector Modulation.....	40
10.3	Interface HTML DSP Control .....	40
10.4	Remarque CPLD Poetic.....	40
11	Simulation Typhoon HIL (Source DC + ANPC + Injection Réseau) .....	41
11.1	Schéma.....	41
11.2	Contrôle de l'installation.....	42
11.3	Communication avec Poetic .....	42
11.4	Panel de commande Typhoon HIL.....	44
12	POETIC + Simulation Typhoon .....	44
12.1	Contrôle du temps d'échantillonnage Typhoon .....	45
12.2	Contrôle de la conversion des signaux fibres optiques .....	45
12.3	Test installation Typhoon .....	47
12.3.1	Open-Loop .....	47
12.3.2	Closed Loop (Grid off) .....	49
12.3.3	Closed-Loop (Grid On) .....	50
12.4	Closed-Loop sur le réseau avec une consigne à 0A.....	51
12.5	Simulation Space Vector Modulation .....	52
12.6	Equilibrage de la tension DC .....	53
12.7	Test de performance de l'installation .....	55
12.7.1	Fréquence PLL.....	55
12.7.2	Vitesse de régulation du courant.....	56
12.7.3	Tension DC limite .....	56
12.7.4	Test FFT des courants pour les harmoniques .....	57
13	Test sur l'installation réelle charge RL .....	60
13.1	Test à une tension d'UDC = 60V .....	60
13.1.1	Test de tension en sortie .....	60

13.1.2	Test de courant avec charge .....	62
13.1.3	Puissances actives et réactives.....	64
13.1.4	Test de fréquence .....	65
13.2	Test à une tension d'UDC = 120V .....	66
13.2.1	Test de courant avec charge .....	66
13.2.2	Puissances actives et réactives.....	66
13.2.3	Test de fréquence .....	67
13.3	Test à une tension d'UDC = 200V .....	68
13.3.1	Test de courant en charge .....	68
13.3.2	Test de fréquence .....	69
13.4	Test d'équilibre des tensions DC .....	70
13.5	Test à une tension de 600V .....	71
13.6	Résumé des tests sur l'installation réelle avec des charges RL .....	71
14	Expérimentation avec réseau industriel, mesure et validation. ....	71
15	Problèmes sur l'installation réelle avec le réseau connecté .....	72
16	Remarque .....	74
17	Conclusion .....	74
18	Références.....	75
19	Annexes .....	75
19.1	Quels états sont dangereux ? .....	76
19.2	Protection de la state machine.....	77
19.3	Images régulation PWM.....	81
19.4	Calcul Space Vector Modulation .....	85
19.5	Vecteurs à choisir pour SVPWM.....	86
19.6	Comparaison Matlab entre version SVPWM_1 et SVPWM_2 .....	87
19.7	Branchement .....	89
19.7.1	Poetic + Typhoon .....	90
19.7.2	Poetic + Système réel .....	93
19.8	Machine d'état du système .....	95
19.9	Schéma du Code Poetic .....	96
19.10	Code Space vector modulation.....	97
19.11	Résultat Space Vector Modulation spécial.....	104
19.12	Transmission des données .....	105
19.13	Mode d'emploi pour activer l'installation .....	106
19.13.1	Poetic + Typhoon .....	106
19.13.2	Poetic + Système réel .....	107

## 1 Remerciements

Merci à mon Responsable du TD, M. Kenzelmann pour le suivi et l'aide tout au long du projet.

Merci au Responsable du Laboratoire M. Germanier pour les explications de la partie Poetic.

Merci à M. Barrade pour l'aide avec la plateforme de simulation Typhoon HIL.

## 2 Introduction

Ce travail de Bachelor consiste à développer le contrôle d'un système onduleur ANPC (Active Neutral-Point Clamped) pour y raccorder des panneaux photovoltaïques ou d'autres sources DC. La technologie ANPC est un onduleur avec un neutre actif à trois niveaux. Un onduleur trois niveau est généralement avec une topologie NPC qui est plus simple mais n'est pas aussi polyvalente.

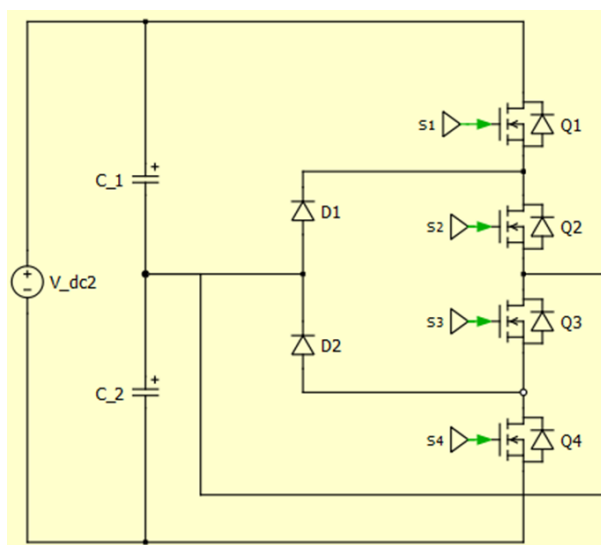


Figure 1 : Topologie NPC

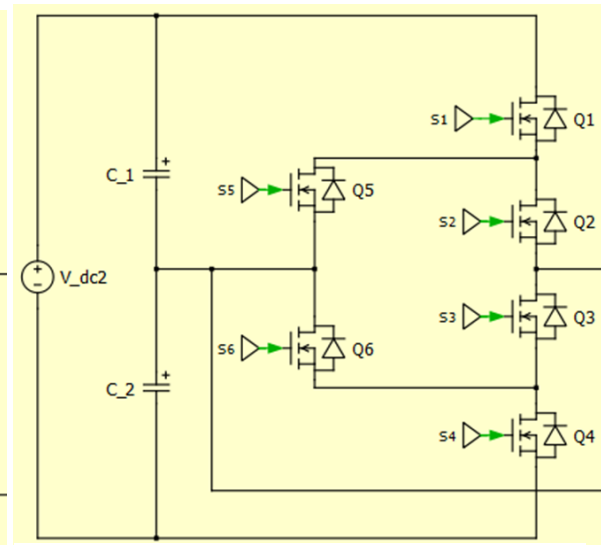


Figure 2 : Topologie ANPC

Un onduleur 3 niveaux peut produire une tension positive P, négative N et un 0V en sortie. Dans la méthode NPC le 0V est créé en passant par les deux 'chemins' avec D1 et Q2 ou D2 et Q3. Les semi-conducteurs Q2 et Q3 sont obligatoirement fermés en même temps pour créer le 0V avec un courant dans les deux sens.

Mais pour la topologie ANPC le 0V peut être produit en choisissant uniquement un de ces deux chemins, car les diodes sont remplacées par des semi-conducteurs. Soit avec Q2 et Q5 ou Q3 et Q6. Ceci permet d'utiliser un seul chemin pour un courant dans les deux sens et en cas de défaut il suffit de choisir l'autre option et ceci permet d'éviter le chemin défectueux en gardant une partie du fonctionnement.

Ainsi avec le système ANPC il est possible de garder le système activé même en cas de défaut à une puissance réduite mais aussi un contrôle plus important du fonctionnement sans défaut.

Le système général correspond à ce schéma (Figure 3). Un panneau photovoltaïque avec une tension DC, suivi d'un onduleur qui se connecte au réseau.

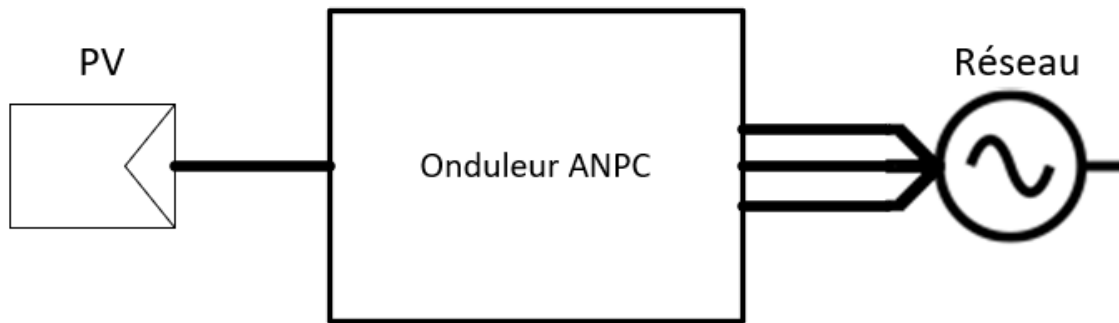


Figure 3 : Schéma simple avec l'onduleur ANPC

L'onduleur ANPC de la Figure 3 contient plusieurs parties. La partie Poetic, qui correspond au cerveau de l'installation. Elle récupère les mesures des cartes de puissances et selon le fonctionnement voulu de l'installation. Elle transmet des données de modulation aux trois cartes de puissances pour les trois phases en passant par une CPLD back-plane qui transforme les signaux électriques en signaux fibres optiques.

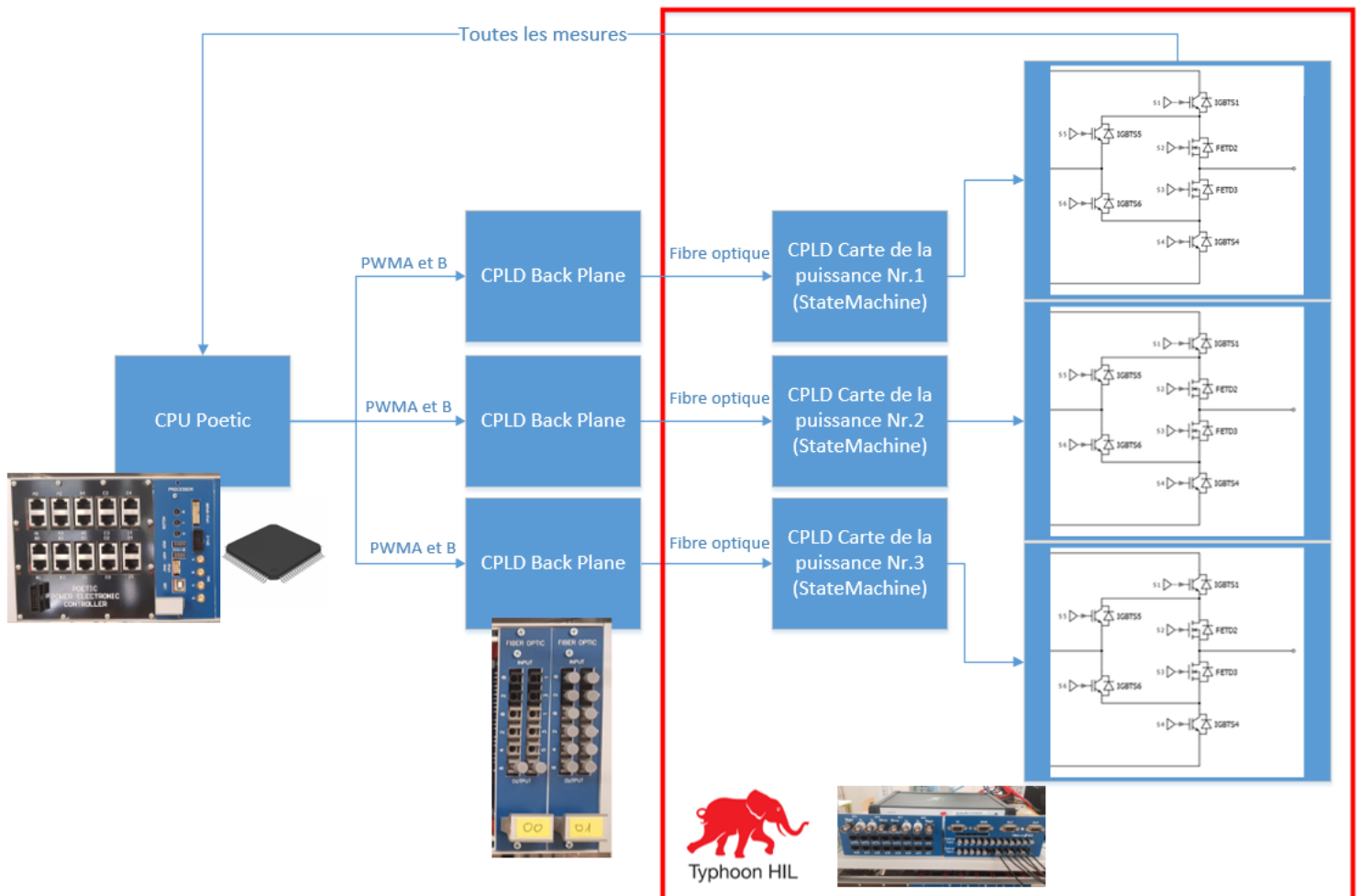
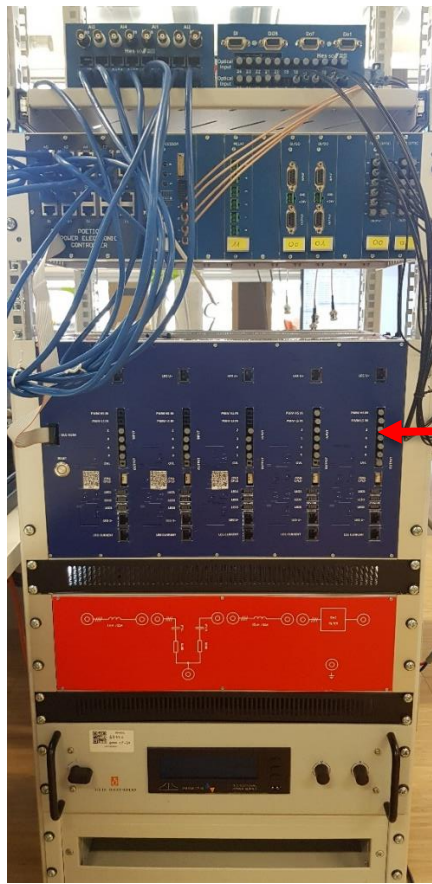


Figure 4 : Généralisation du Système avec des images, Cette figure est utilisée pour aider à la compréhension au début de chaque chapitre



## 2.1 Bases du système



← Typhoon HIL

← Poetic (CPU, fibres optiques et mesures par Ethernet)

← Cartes de puissances

← Filtre en sortie du système

← Alimentation DC

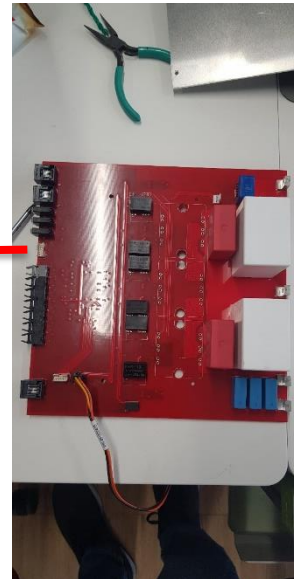


Figure 5 : Rack avec toutes les parties du système. L'image de droite est l'une des cartes de puissances avec une CPLD qui permet le contrôle des semi-conducteurs. Ces cartes sont montées dans le rack (3 cartes de puissance pour un système triphasé)

Poetic est un système qui permet de rapidement faire des modifications de code et branchements. Poetic contient un CPU qui gère le code. Il est connecté à une back-plane. La back-plane permet d'y brancher des modules différents comme des entrées ou sorties de tout type. Dans ce projet sur cette back-plane il y a les fibres optiques en entrées et sorties qui y sont branchées. Ces fibres optiques envoient ou reçoivent les données des cartes de puissance.

Il y a également une partie mesure. Celle-ci permet de brancher jusqu'à 20 mesures maximum pour contrôler le système avec des câbles Ethernet.

Sur la Figure 6, de gauche à droite. Il y a les mesures avec les prises Ethernet suivi du CPU qui gère le fonctionnement de l'installation. Les modules à droite du CPU peuvent être changés selon les besoins. Pour le système ANPC les seuls modules utilisés sont les fibres optiques.



Figure 6 : Système POETIC



Les signaux fibres optiques se branchent sur les cartes de puissances. Il y a l'emplacement pour mettre 5 cartes mais uniquement 3 sont utilisées pour les 3 phases du système.



Figure 7 : Face avant des cartes de puissances avec les branchements fibres optiques et mesures

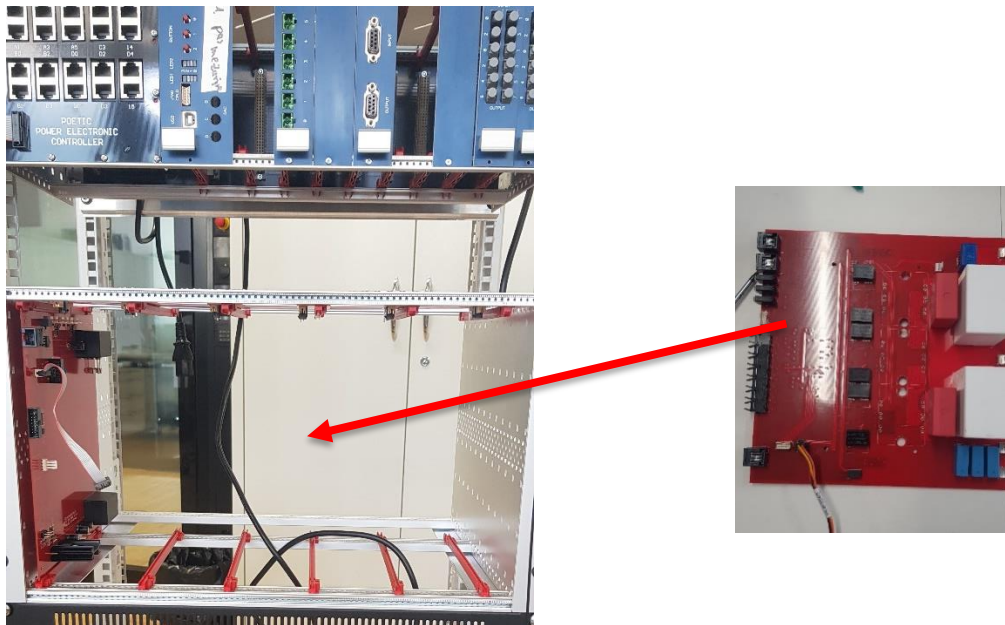


Figure 8 : Emplacement pour mettre les cartes de puissances dans le rack

Pour les tests du système il est faut utiliser Typhoon (Figure 9). Typhoon est un programme qui permet de faire de la simulation temps réel avec des entrées et sorties. Dans ce projet les cartes de puissances sont simulées dans le programme et vont réagir aux signaux reçus et fonctionner comme les cartes réelles.

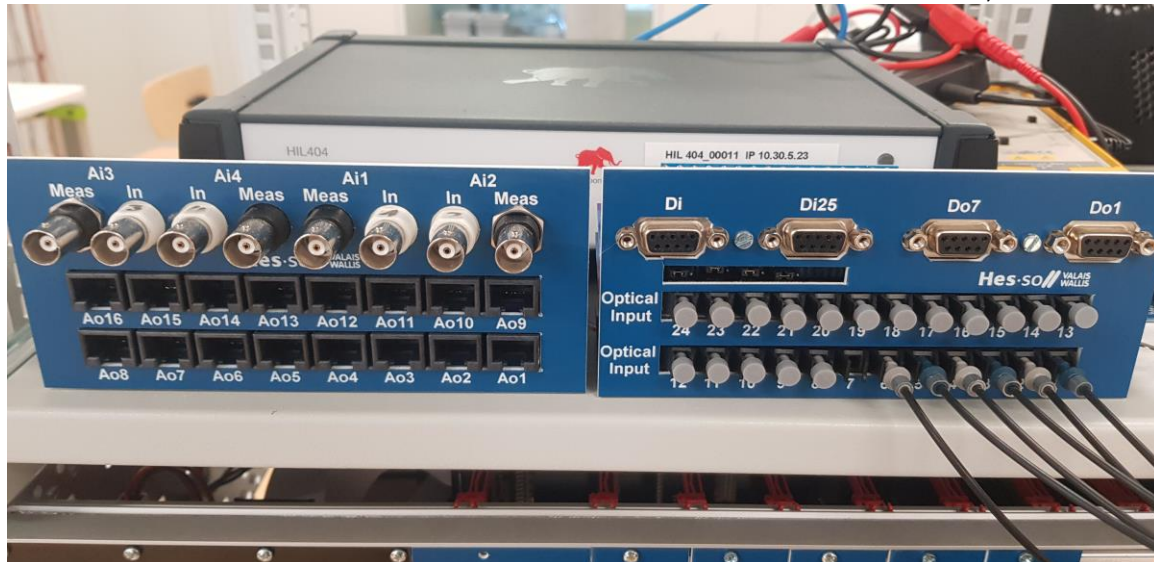


Figure 9 : Typhoon HIL avec des entrées et sorties Ethernet pour les mesures ainsi que des entrées et sorties fibres optiques

### 3 But et Objectif

Le but de ce travail de Bachelor est de réaliser le système Onduleur ANPC qui est alimenté par des panneaux photovoltaïques ou d'autres alimentations DC pour injecter du courant dans le réseau ou pour faire tourner un moteur.

En début de projet, il est important de lire les papiers de recherche à disposition sur cette méthode ANPC pour comprendre le bon fonctionnement du système. Avec sa state machine (chapitre 5.1) pour les semi-conducteurs, ainsi que les méthodes de pilotage (PWM au chapitre 7.1 ou SVPWM au chapitre 7.2) et les avantages et inconvénients des différentes méthodes de pilotage ou les différents types de semi-conducteurs, ainsi que ses performances.

La protection du système est très importante. Les drivers des semi-conducteurs sur les cartes de puissances fournissent des signaux Desat, qui correspondent à un défaut sur un des semi-conducteurs.

Si celui-ci s'active pendant que l'installation est en marche il faut de manière sûre désactiver la phase en défaut. Il est également possible de bloquer la phase à la tension de 0V et continuer à puissance réduite.

Une fois qu'on a une vue globale du fonctionnement de la topologie, il est possible de faire des simulations sur PLECS (PLECS est un programme de simulation de système). Ces simulations permettent de contrôler la régulation ainsi que les state machines et voir le fonctionnement. Il est aussi possible de déterminer les pertes de conduction et de commutation et faire des comparaisons de performances pour différents semi-conducteurs. (Choix entre IGBT, Mosfet ou Hybride).

Lorsque les simulations sont fonctionnelles il faut programmer le système de contrôle sur le CPU Poetic en code C (Chapitre 10) Le CPU contient la régulation et le fonctionnement de l'installation.

Afin de tester le système il faut avancer par étape en remplaçant des parties du système par une simulation Typhoon HIL (Chapitre 11) et après confirmation que le système est opérationnel avec Typhoon il sera possible de tester l'installation avec les cartes de puissances réelles.

Pour tous ces tests, il faut prendre des mesures et ensuite comparer ces valeurs avec les simulations ou comparer entre les différents semi-conducteurs (Mosfet ou IGBT).

### 3.1 Les étapes du projet

Rouge = Rien à faire, Orange = Modification spécial sur CPLD, Vert = A faire

Titre avec chapitre du rapport	Description
Commande de matériel	Déjà disponible
Préparation des cartes électronique	Déjà faites
Câblage électrique du système	Une autre personne du projet ANPC fait le câblage
Compréhension de la topologie ANPC (0)	Lecture des papiers de recherche [1][2][4]
Description de la state machine pour les cartes de puissance (5)	Description du fonctionnement avec tous les états et le code sur les CPLD des cartes de puissances est fait par quelqu'un d'autre
Simulation PLECS (8)	Simuler le fonctionnement avec la régulation
Programmation Poetic CPU (10)	Des bases existent pour le fonctionnement de POETIC, mais la partie régulation et Space Vector modulation est à faire et quelques modifications
Programmation Poetic CPLD (10.4)	Modification pour adapter le système ANCP sur la CPLD (sorties PWMA et PWMB se bloquent)
Simulation Typhoon (11)	Créer la structure pour simuler le système avec Typhoon
Câblage des mesures et fibres optiques (19.7 en annexe)	Câblage sur Typhoon ou l'installation réel selon les tests nécessaires
Test sur Typhoon (12)	Tester le fonctionnement en remplaçant les alimentations et cartes de puissances réel par Typhoon
Test réel (13)	Tester le fonctionnement sur les cartes de puissances réel

Tableau 1 : Etapes à faire pour le projet avec les références

## 4 Recherche littérature

Pour commencer dans ce projet ANPC il faut tout d'abord comprendre le fonctionnement de cette topologie d'onduleur. Il y a plusieurs papiers de recherche qui explique le fonctionnement et les différentes manières de piloter le système. Le premier papier de recherche est celui qui a été le plus important avec le plus d'information de base [1] (UnitedSiC). Il explique les différences entre NPC et ANPC ainsi que des comparaisons de performances. Les pertes entre différentes manières de piloter le système et les pertes selon les composants choisis (Mosfet ou IGBT). Mais aussi un chapitre sur le fonctionnement normal et les états potentiellement dangereux. Ces états sont à éviter et considéré comme état interdit pour protéger le système. Exemple : Il est interdit d'activer les composants S1, S2 et S3 en même temps, car cela peut détruire des composants et ainsi le système (Figure 10). Il y a plusieurs de ces cas à éviter. (Tableau avec tous ces états dangereux en annexe au chapitre 19.1)

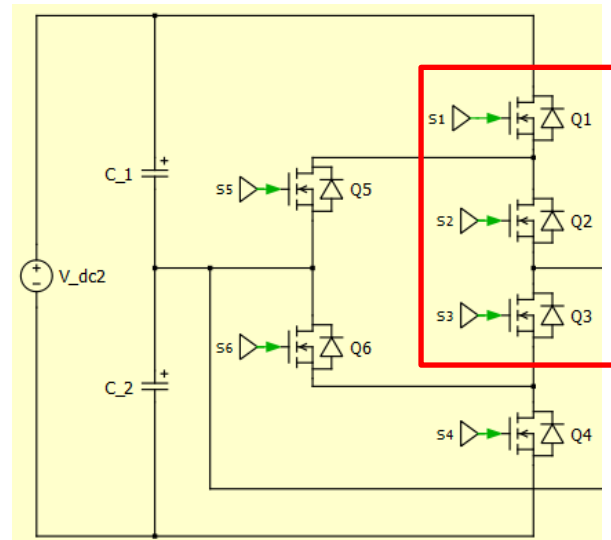


Figure 10 : Etat dangereux en fonctionnement, S1, S2 et S3 ne peuvent pas être activé en même temps, état interdit

Une autre partie importante de ce papier [1] est la manière de piloter le système. Il y a plusieurs state-machines décrites avec une comparaison de ses rendements. La solution avec le meilleur rendement correspond à la state machine PWM4. C'est cette méthode qui est utilisé dans ce projet pour le fonctionnement du système. L'image de la state machine est disponible à la Figure 13.

Le papier de recherche [2] est pour la commande en Space Vector Modulation de l'onduleur et explique son fonctionnement.

Le [3] est un rapport de travail de Bachelor de M. Gabriele Pescatore. Il permet de choisir les bons gains pour les entrées et sorties de la simulation Typhoon.

Le [4] correspond au papier de recherche qui traite du fonctionnement en cas de défaut sur une phase, suivi d'un pilotage du système sur deux phases en bloquant la dernière à 0V.

## 5 State machine des cartes de puissance

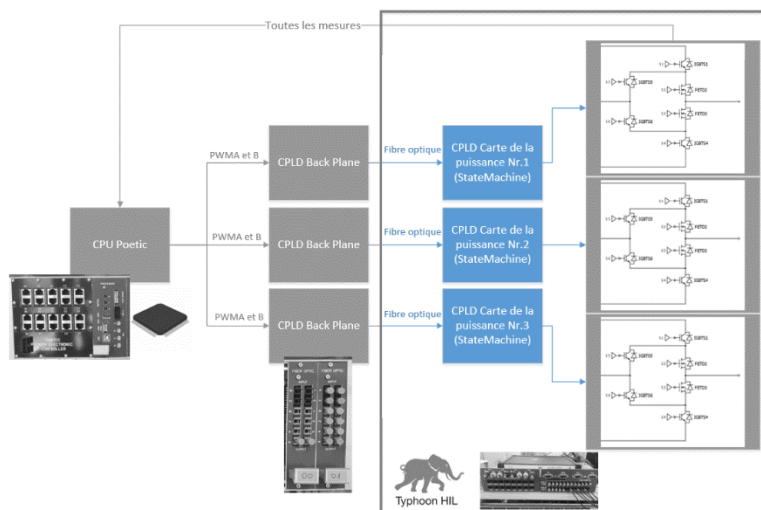


Figure 11 : Ce chapitre parle de la partie qui est en couleur, CPLD carte de puissance

Chaque carte de puissance possède sa propre state machine. Selon les deux signaux fibres optiques reçu la state machine va changer d'état.



### 5.1 Explications State Machine des cartes de puissance

Le système produit 3 états en sortie. Le P qui correspond à une tension positive en sortie, Le N qui est une tension négative en sortie, le O qui met la tension à 0V en sortie et l'état OFF qui ouvre tous les semi-conducteurs et coupe l'installation. Les deux autres états PO et NO ne sont pas utilisés pour piloter l'installation, ils sont uniquement des états de passage (PO et NO donnent une tension de 0V). Pour passer d'un état à un autre il faut attendre une durée Ton ou Toff. Ces temps correspondent à la durée pour être sûr que les semi-conducteurs ont eu le temps de se fermer ou s'ouvrir. (Ces valeurs dépendent du composant et c'est disponible à la Figure 14.)

Il y a différents états possibles (En lien avec la Figure 12) :

P : Activation Cycle Positif  
 P  $\leftrightarrow$  O : Intermédiaire Cycle Positif  
 O : Neutre (0V en sortie)  
 N  $\leftrightarrow$  O : Intermédiaire Cycle Négatif  
 N : Activation cycle Négatif  
 OFF : Eteint, Tous les semi-conducteurs sont ouverts

La commande pour passer d'un état à l'autre correspond à la tension voulu en sortie. Pour une tension positive la machine d'état passe de O vers le P en passant par l'état intermédiaire (Etat intermédiaire pendant une courte durée).

Pour la tension négative la machine d'état passe de O vers le N en passant par l'état intermédiaire. Pour une tension voulu de 0V en sortie le système reste dans l'état O.

Si une tension plus faible que la tension P est nécessaire en sortie il faut faire de la modulation. Exemple : Pour obtenir la moitié de la tension P en sortie, c'est possible en restant uniquement 50% du temps dans l'état P et le reste du temps en 'redescendant' à l'état O. Si la modulation est rapide il est possible de passer de l'état P vers l'état intermédiaire et repasser dans l'état P sans passer par l'état O. Les états intermédiaires sont également à 0V et ne pas passer par l'état O n'est pas un problème.

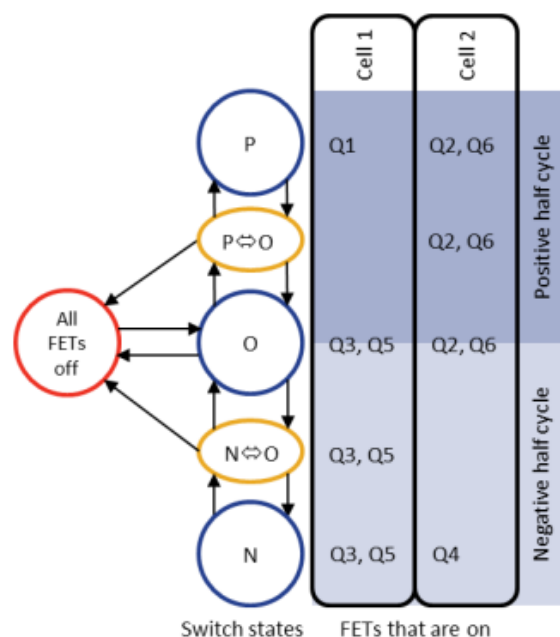


Figure 12 : Schéma UnitedSic [1] pour le fonctionnement PWM4 (state machine des cartes de puissances)

## 5.2 Explication du signal Desat

Tous les semi-conducteurs possèdent un gate-driver qui le pilote et sur ce driver il y a quelques sorties. Une de ces sorties est le Desat qui indique qu'un court-circuit sur le semi-conducteur est apparu et le système peut choisir ce qu'il faut faire. Il y a également un signal Ready qui protège contre des défauts moins urgents. Exemple : Si la température du composant est trop importante, le signal Ready se met à 0 et coupe le semi-conducteur. Ce n'est pas un danger immédiat mais permet de protéger les semi-conducteurs et ne pas fonctionner aux limites possibles de l'installation et préserve la durée de vie du système. Ces signaux sont traités directement avec la CPLD de la carte de puissance. Mais il faut quand même avertir le CPU du problème.

Ces deux signaux communiquent avec le CPU en coupant le signal de la fibre OVL. OVL est prévu pour protéger des sur-courants mais ces deux signaux y sont également connectés.

En cas de défaut il faut rapidement couper le système ou le mettre à 0V. En mettant à 0V il est possible de continuer à fonctionner à puissance réduite, car la topologie ANPC le permet. Cependant il n'est pas possible d'éteindre tous les composants en même temps, car l'ordre de coupure est aléatoire. Pour certain cas vu dans le chapitre 0, il y a des états interdit. L'ordre aléatoire peut engendrer ces états interdits/dangereux. Ainsi il faut suivre un ordre très précis d'état pour couper l'installation en cas de défaut sur un composant. Dans le chapitre 5.3 la démarche à suivre en cas de défaut pour chaque état est décrite ainsi qu'un exemple.

## 5.3 Démarche en cas de défaut

Pour protéger le système en cas de défaut sur n'importe quels semi-conducteurs, il faut suivre une procédure pour éteindre les bons semi-conducteurs dans un ordre précis. L'image ci-dessous correspond à la machine d'état de la Figure 12. En ajoutant des états à cette state machine il est possible de protéger le système dans n'importe quel état.

Machine d'état PWM4 de base sans les protections. La PWM4 avec les protections est à la Figure 16

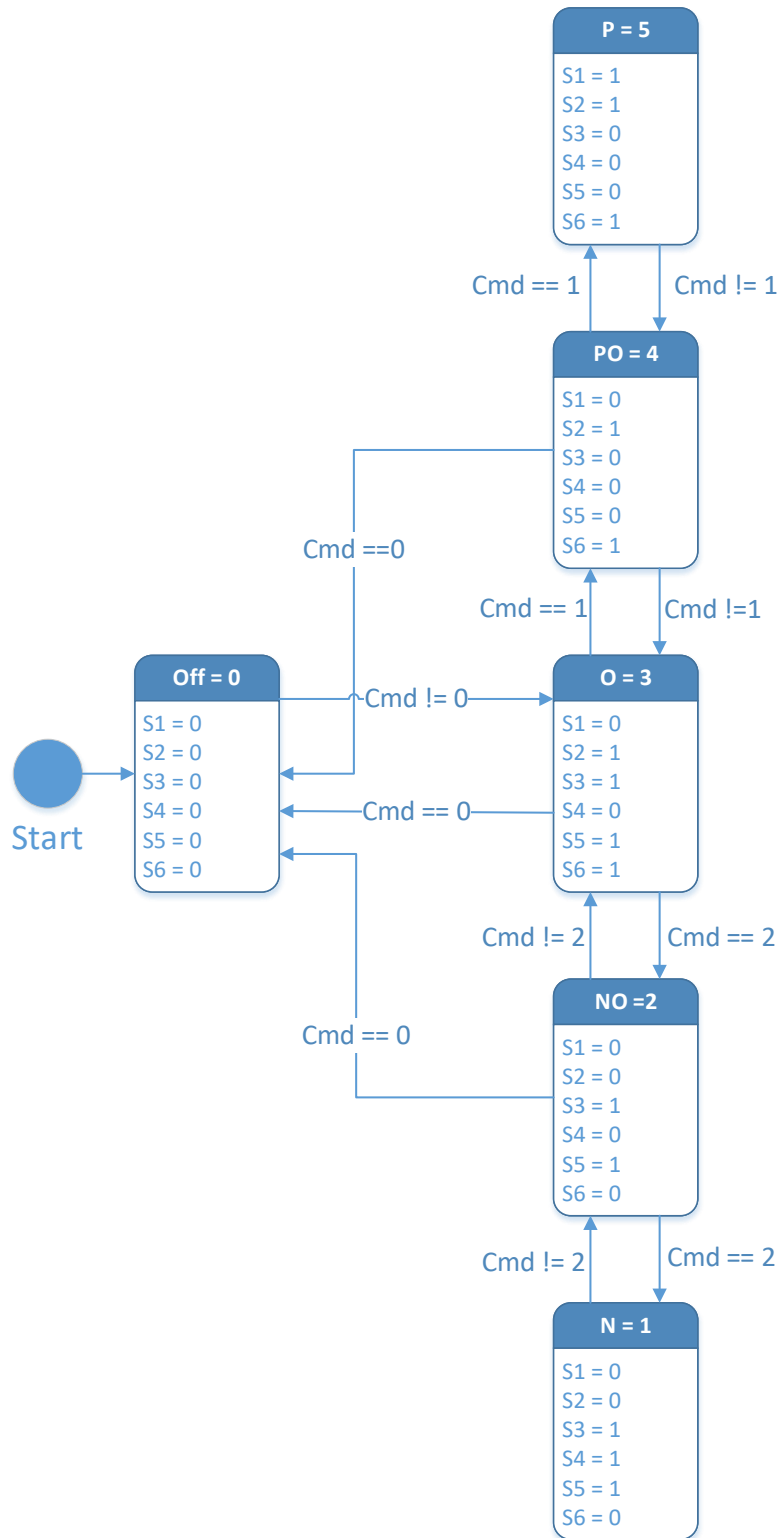


Figure 13 : Fonctionnement PWM4 avec la state machine sans protection desat, (cmd va de 0 à 3)

Pour les cas de défauts il faut ajouter des états à la state machine et selon le Desat se déplacer dans des états spécifiques au problème.

Les Figure 14 et Figure 15 permettent de comprendre comment le schéma est construit avec des explications :

(Ces schémas avec plus de détails et explications sont disponibles en annexe 19.2)

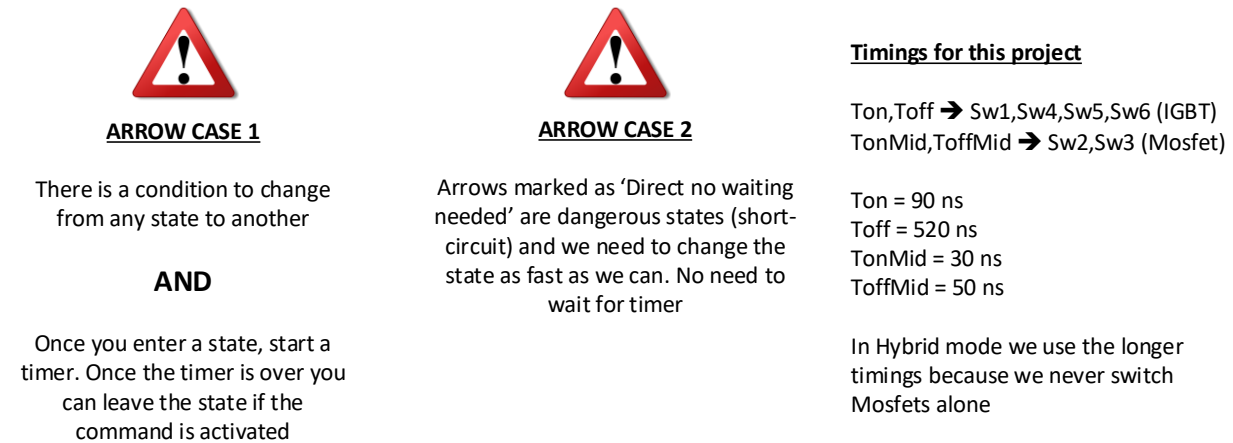


Figure 14 : Warning sur le fonctionnement de protection et des transitions avec les temporisations

### Example and explanations

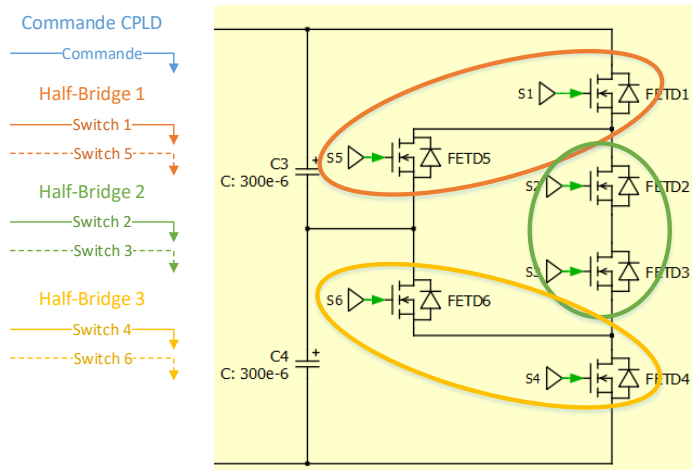
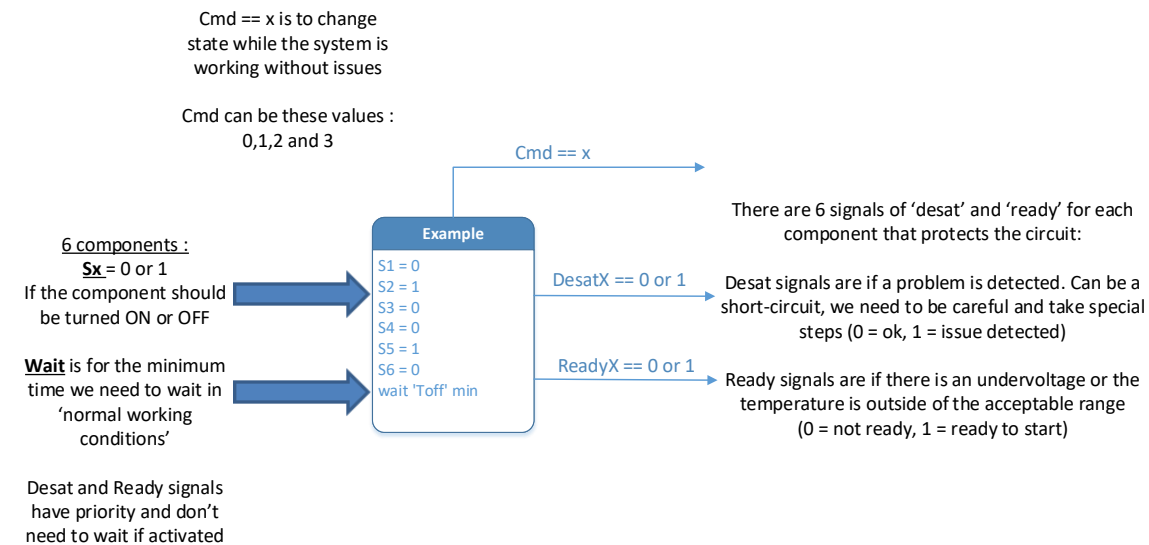


Figure 15 : Schéma des composants avec dénomination pour la compréhension de la machine d'état



La machine d'état de base en fonctionnement sans défaut est encore visible sur la droite de l'image. Mais beaucoup d'états supplémentaires ont dû être ajoutés pour la protection.

Vue générale : (Pour voir en détail ce qui est écrit il y a l'image disponible en annexe 19.2)

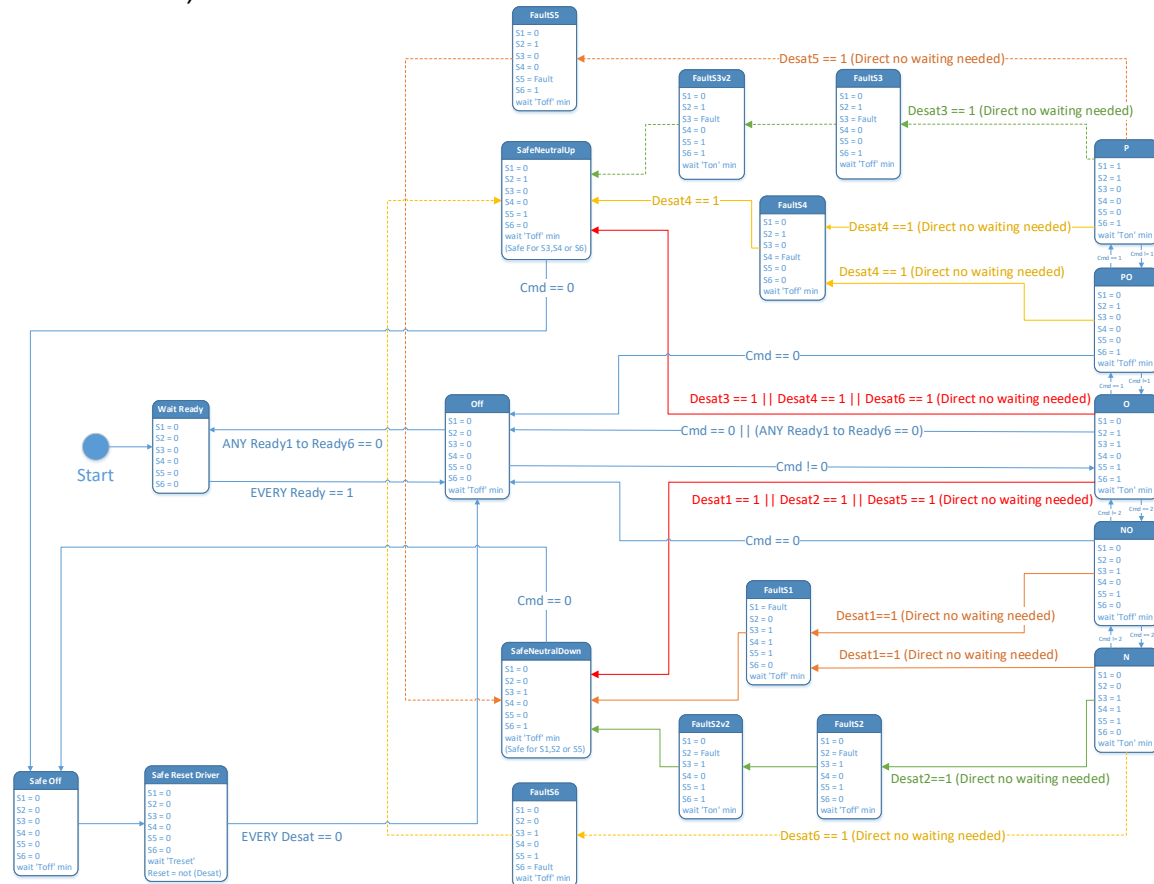


Figure 16 : Fonctionnement de protection dans la machine d'état

Exemple de protection : Figure 17. Voici l'état du système lorsqu'il est en mode P avec S1, S2 et S6 fermés. S'il y a un court-circuit sur le Switch S3, cela correspond à un état interdit avec S1, S2 et S3 fermés. Il faut donc rapidement ouvrir un de ces composants pour ne pas endommager le circuit.

Pour tous les cas possibles il faut avoir une solution avec des états spécifiques. (Quelques états sont les mêmes).

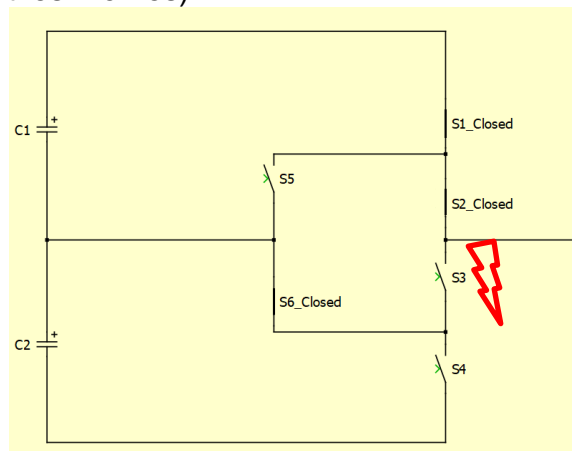


Figure 17 : Schéma de représentation d'un court-circuit sur S3 dans l'état P de la state machine

## 6 Transmission des paramètres de modulation vers l'onduleur

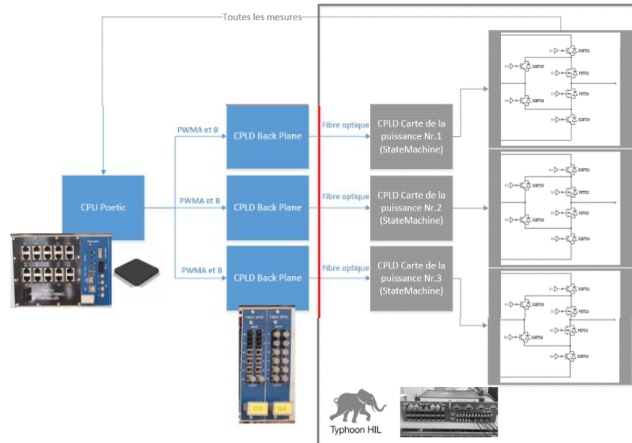


Figure 18 : Ce chapitre traite des parties du système en couleur

La transmission des signaux ne dépend pas du choix de fonctionnement de la modulation. PWM (chapitre 7.1) et SVPWM (chapitre 7.2) utilisent le modulateur. La transmission se fait à l'aide du modulateur. Les paramètres de modulation de l'installation sont envoyés dans le modulateur et c'est le modulateur qui transmet directement les données par les fibres optiques PWMx\_A et B.

### 6.1 Modulateur

Le modulateur utilise une valeur de modulation créée par une des deux méthodes, soit PWM ou SVPWM. Cette valeur de modulation est comparée avec une carrier frequency. Elle varie de 0 à 2'500 et redescend à 0 en une seule période. Donc 5'000 ticks par période à 100MHz ce qui donne une période de 50us ou 20kHz.

Les registres à modifier :

TBPRD pour la période

AQCTLA et AQCTLB pour la logique sur la sortie (Inversion du signal)

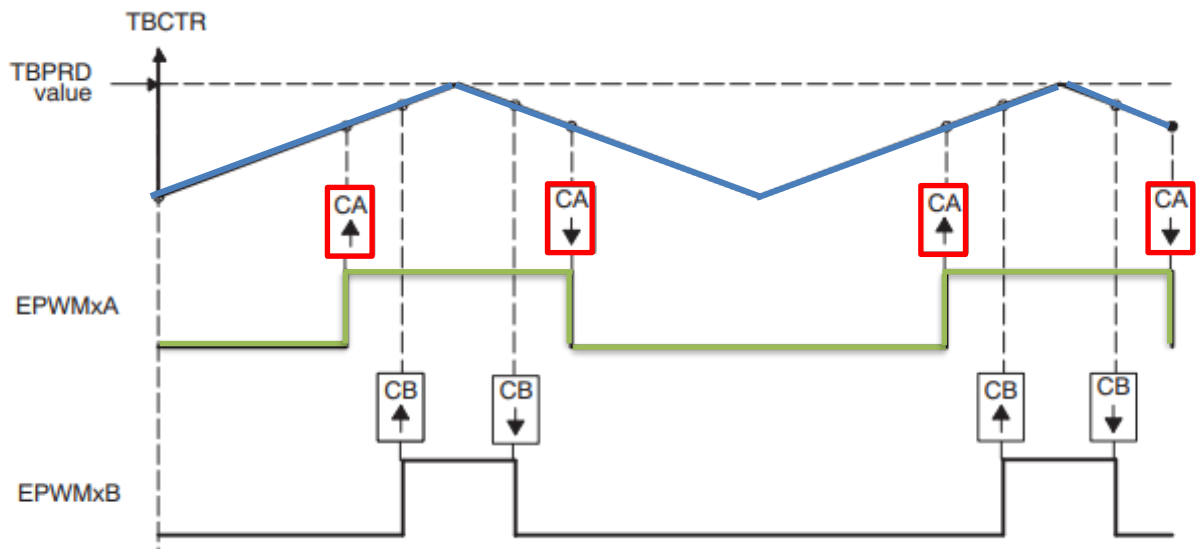


Figure 19 : Fonctionnement du modulateur. Le signal bleu est la carrier frequency. Les points rouges sont les valeurs de la modulation. Le signal vert est la sortie du modulateur

Le signal rouge est le signal de modulation et il varie à 50Hz ce qui est très faible comparé aux signal bleu qui correspond à la carrier frequency (400 fois plus rapide que la modulation). La modulation est donc 'fixe' entre le début et la fin de chaque période de la carrier frequency.

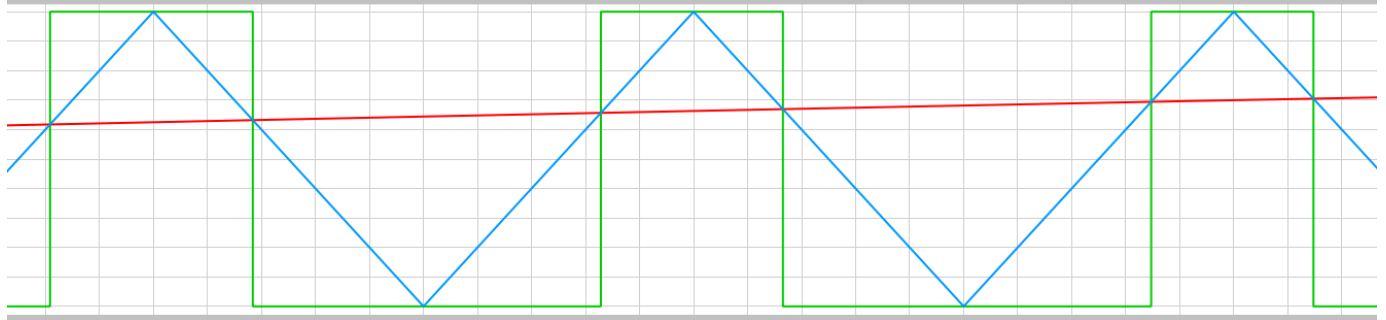


Figure 20 : Correspond aux simulations (Plecs) → Carrier frequency (bleu) avec signal de modulation (rouge) et la sortie (vert) varie en comparant les deux valeurs

## 6.2 Signaux pour les modulateurs PWMA et PWMB

Les valeurs à mettre sur les modulateurs sont en sortie de la régulation du système. Ils correspondent à des tensions à appliquer sur les phases de l'onduleur. Ces signaux sont des sinus. Mais pour les utiliser il faut premièrement les normaliser avec la tension DC pour avoir des valeurs entre (-1 et 1)

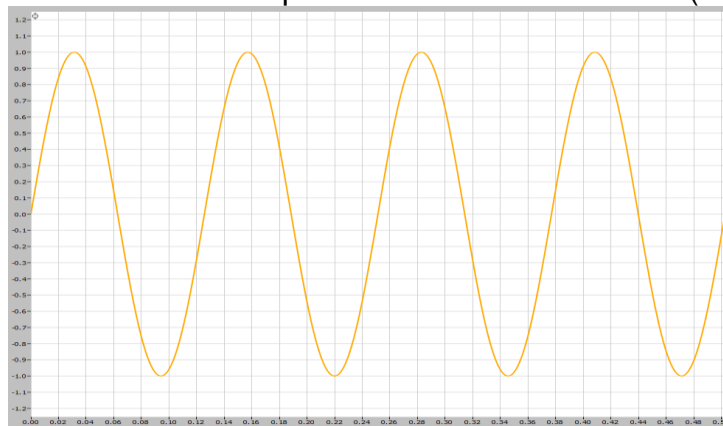


Figure 21 : Exemple de signal normalisé pour aller sur un modulateur (valeur entre -1 et 1)

Pour envoyer ce signal sur le modulateur il faut séparer le signal entre la partie positive et négative :

Le plus simple est de créer deux signaux.

Le premier signal ne change pas.

Le deuxième doit être inversé → Multiplié par (-1)

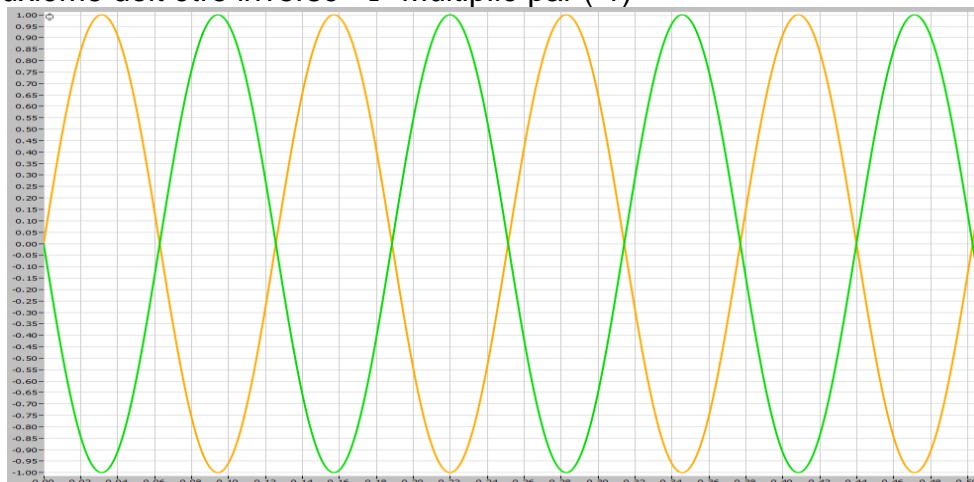


Figure 22 : Signal de modulation séparé pour le modulateur A et B

Ensuite il faut uniquement garder la partie positives de ces deux signaux.

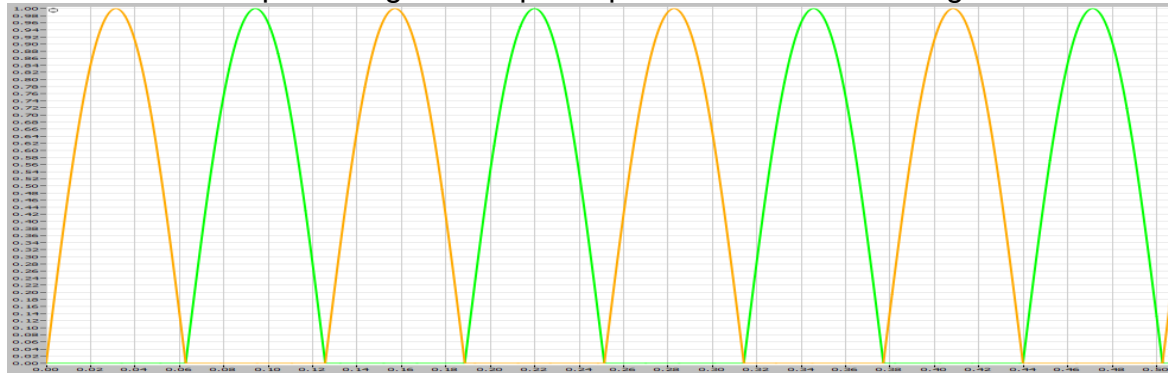


Figure 23 : Limite la valeur des signaux. Garde uniquement la partie positives

Ce signal est prêt pour être envoyés sur les modulateurs.

Le premier signal sur le modulateur A pour la sortie PWM\_A et le deuxième sur le modulateur B pour le PWM\_B (celui qui a le gain de -1).

(L'étape de limitation du signal n'est pas obligatoire, car le modulateur limite déjà la valeur de 0 à 1 à l'acquisition des signaux, mais cela permet de mieux comprendre le fonctionnement de la normalisation et adaptation des signaux avant de l'envoyer dans une autre fonction).

### 6.3 Transmission des données de modulation vers l'onduleur

L'installation à disposition fonctionne normalement pour un onduleur 2 niveaux qui a besoin d'uniquement savoir s'il faut activer ou éteindre les deux semi-conducteurs du système. La commande avec les fibres optiques indique aux semi-conducteurs directement s'il faut être ouvert ou fermé. Dans un onduleur 3 niveaux il y a 4 états de fonctionnement : +1, -1, 0 et état OFF. En gardant un système avec deux fibres optiques il faut changer la méthode de transmission des données, car il y a plus que deux semi-conducteurs et une commande directe n'est plus possible. En utilisant ces 2 fibres optiques pour représenter les 4 états de l'onduleur 3 niveaux (6 Fibres optiques en tout pour les 3 phases) avec de la logique binaire il est possible de 'coder' les signaux à envoyer et les signaux reçus, ce qui donne 4 états différents possible. La méthode est la suivante :

Les valeurs pour la PWM 3 niveaux sont 1, -1 et 0 qui sont transformé en signal binaire de 1, 2 et 3 respectivement. Ces valeurs sont codées sur les deux fibres optiques comme le tableau suivant et au chapitre 6.4:

Valeur PWM 3 niveaux	Signaux PWM en sortie	Fibre Optique (FO01)	Fibre Optique (FO02)
OFF	0	0	0
1 (High)	1	0	1
-1 (Low)	2	1	0
0 (Middle)	3	1	1

Tableau 2 : Tableau qui fait le lien entre la tension voulu en sortie avec le signal codé et les valeurs sur les fibres optiques

Ainsi il est possible d'envoyer les informations du fonctionnement avec ses 4 états pour la commande avec ces deux fibres optiques.

### 6.4 Transmission des données, Cas Concret

Le signal sinus pour faire de la modulation est séparé en deux parties pour être envoyé vers les deux PWM. PWM\_A pour la partie positive et PWM\_B pour la partie négative.

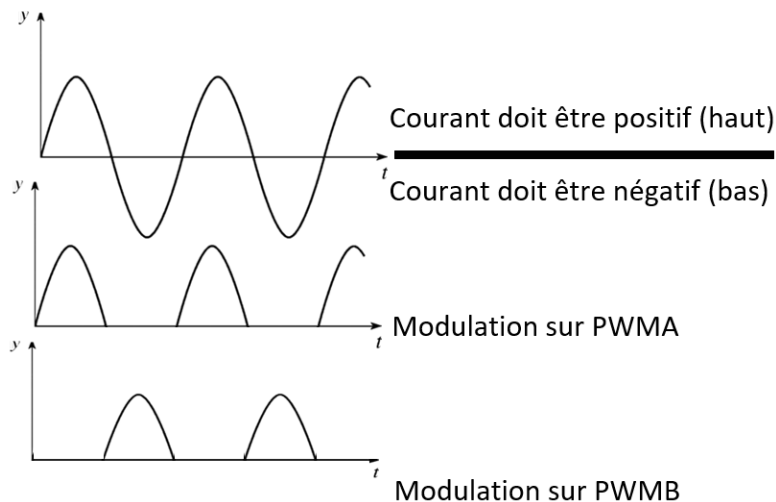


Figure 24 : Le signal sinus est séparé en deux parties pour le PWMA et PWMB

Les sorties PWM\_A et B sont soit à 0 ou à 1.

Lorsque le sinus de référence est à 0, la sortie du modulateur est à 1 pour le PWM\_A et à 1 pour le PWM\_B. (Gain de 2 en sortie du PWM\_A car c'est le MSB du signal binaire)

Le PWM\_A va moduler uniquement lorsque le signal est dans sa partie. Sinon il reste à 1. Même chose pour le PWM\_B.

Les cartes de puissances qui reçoivent les signaux PWM\_A et PWM\_B **font la somme** des deux valeurs en sortie de modulation, ce qui peut donner un signal de 1, 2 et 3 (**Attention gain de 2 sur PWM\_A car c'est le MSB du signal binaire**). Pour avoir 0 il faut couper les PWM.

La somme de ce signal correspond à la commande de la state machine à la Figure 13.

PMW_A (gain de 2, car MSB)	0	0	2	2
PWM_B	0	1	0	1
Somme	0	1	2	3

Tableau 3 : Somme des deux signaux fibres optiques, permet de faire la commande de la state machine

Une somme de 0 indique à l'onduleur d'ouvrir tous les semi-conducteurs.

Une somme de 1 indique à l'onduleur d'être à la tension positive.

Une somme de 2 indique à l'onduleur d'être à la tension négative.

Une somme de 3 indique à l'onduleur d'être à 0V.

Explications avec le schéma à la Figure 26.

**Attention lorsqu'on ne module pas (Modulation à 0) le signal en sortie de PWM est à la sortie haute, pas à la sortie basse !!!**

Dans la première partie du signal sinus on module sur le signal PWMA qui a un gain de 2. La somme des sorties PWMA et PWMB donne soit la valeur P(1) ou O(3).

Dans la deuxième partie du signal sinus on module sur le signal PWMB qui a un gain de 1. La somme des sorties PWMA et PWMB donne soit la valeur N(2) ou O(3).

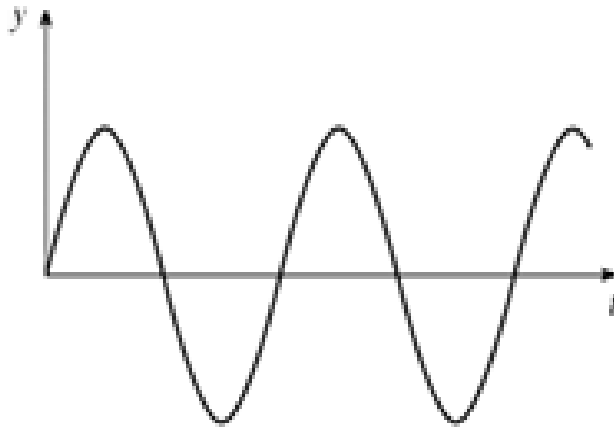
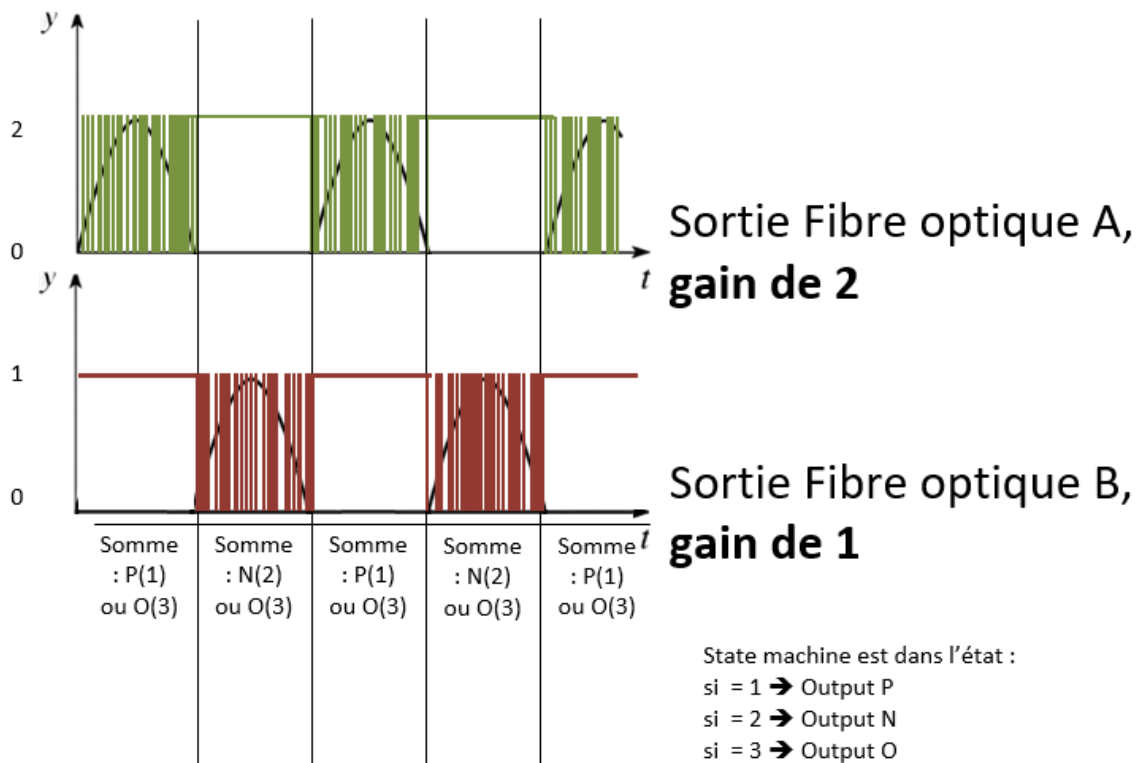


Figure 25 : Sinus de référence pour la modulation



State machine est dans l'état :

si = 1 → Output P

si = 2 → Output N

si = 3 → Output O

Pendant le signal positif on module entre P et O.

Pendant le signal négatif on module entre N et O.

Figure 26 : Sortie des modulateurs, signaux sur la fibre optique qui transmet les bonnes valeurs

## 6.5 Vue en détail des transmissions des données

L'image est disponible en annexe en plus grand au chapitre 19.12.

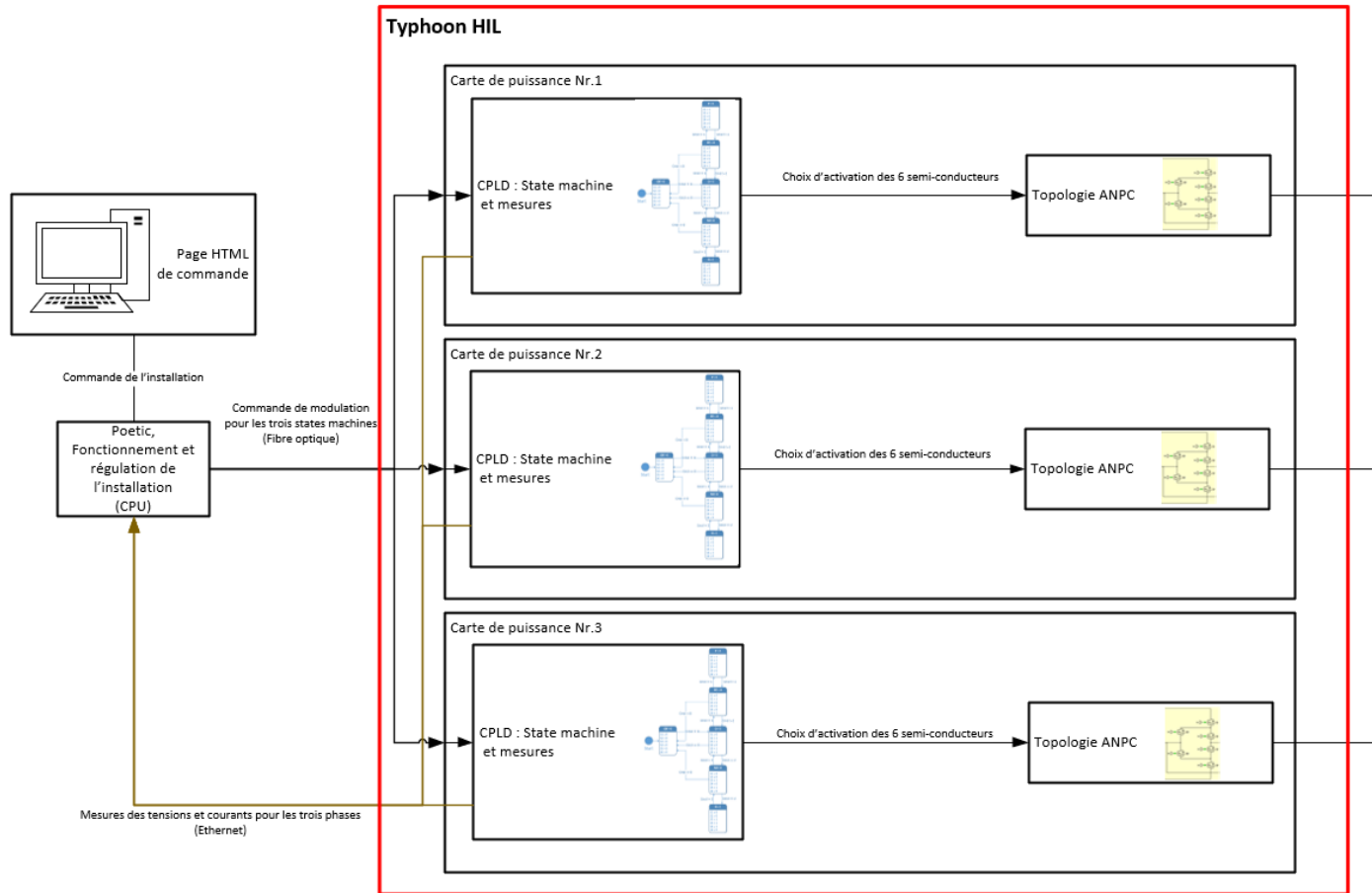


Figure 27 : Schéma du système avec plus de précision

## 7 Contrôle du système

### 7.1 PWM

Cette méthode utilise le modulateur du chapitre 6.1. Il suffit de créer la valeur de modulation avec une régulation. Les valeurs pour les modulateurs sont limitées entre 0 et 1. Pour un signal sinus entre -1 et 1, il faut suivre les étapes du chapitre 6. La régulation est la suivante :

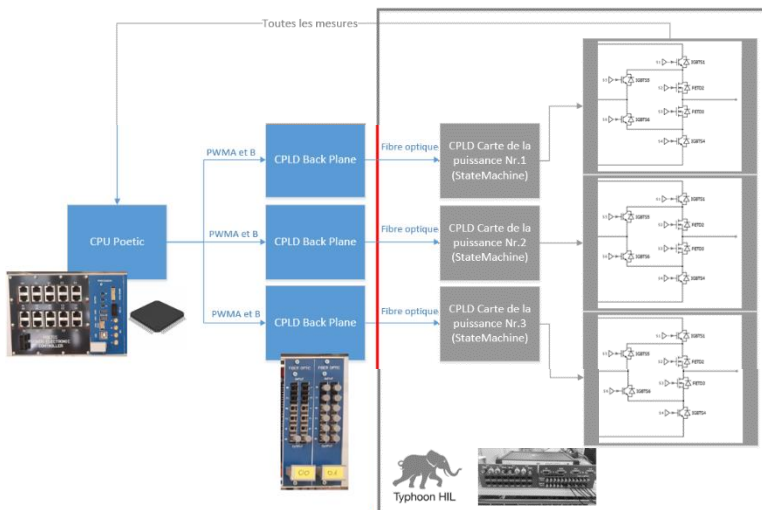


Figure 28 : Ce chapitre traite du contrôle dans le CPU ou la partie en couleur

### 7.1.1 Calcul pour la régulation

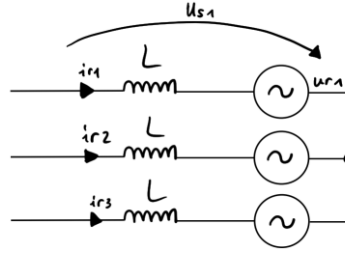


Figure 29 : Schéma pour le calcul de la régulation

$$\begin{aligned}
 u_{s1} &= Reg * i_{s1} + L * \frac{di_{s1}}{dt} + u_{r1} \\
 u_{s\alpha\beta} &= Reg * i_{\alpha\beta} + L * \frac{di_{\alpha\beta}}{dt} + u_{r\alpha\beta} \\
 u_{sdq} &= Reg * i_{dq} + L * \frac{di_{dq}}{dt} * (\cos(\omega t) - j * \sin(\omega t)) + u_{rdq} \\
 \begin{cases} u_{sd} = Reg * i_d + L * \frac{di_d}{dt} - L * \omega * i_q + u_{rd} \\ u_{sq} = Reg * i_q + L * \frac{di_q}{dt} + L * \omega * i_d + u_{rq} \end{cases} \\
 \begin{cases} u_{sd\_ref} = Reg(s)(i_{d\_ref} - i_d) - L * \omega * i_q + u_{rd} \\ u_{sq\_ref} = Reg(s)(i_{q\_ref} - i_q) + L * \omega * i_d + u_{rq} \end{cases}
 \end{aligned}$$

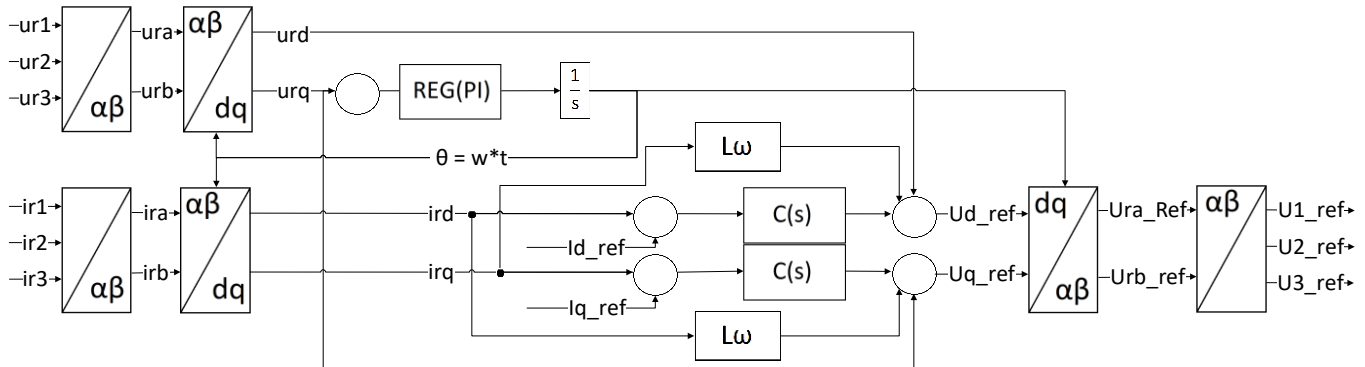


Figure 30 : Régulation du système pour une méthode PWM

La régulation du système permet en fonction des paramètres choisis par l'utilisateur d'adapter le fonctionnement et se retrouver dans les conditions indiquées.

Pour faire la régulation du système plusieurs mesures sont nécessaires. Les trois mesures de tensions et les trois mesures de courant. Ces mesures sont transformées dans le référentiel dq.

La tension Uq permet de calibrer la PLL. La PLL permet de synchroniser le réseau et l'installation ainsi que de connaître la fréquence du réseau, car il est possible que le système ne soit pas connecté à un système 50Hz.

Ensuite la différence entre le courant sur le réseau et la valeur que l'utilisateur veut, correspond à l'erreur pour le régulateur PI. La sortie s'additionne avec un Feed-Forward avec 2 paramètres. La tension Ud ou Uq et aussi le paramètre  $(-I_q * L * 2 * \pi * \text{fréquence})$  sur la sortie du régulateur Id.

Et le paramètre  $(I_d * L * 2 * \pi * \text{fréquence})$  sur la sortie du régulateur Iq. ATTENTION à l'inversion et aux signes !



## 7.2 Space Vector Modulation

La méthode Space Vector Modulation utilise le modulateur du chapitre 6.1. La méthodologie est différente mais une grande partie de la régulation est la même. La base du fonctionnement est l'hexagone SVPWM.

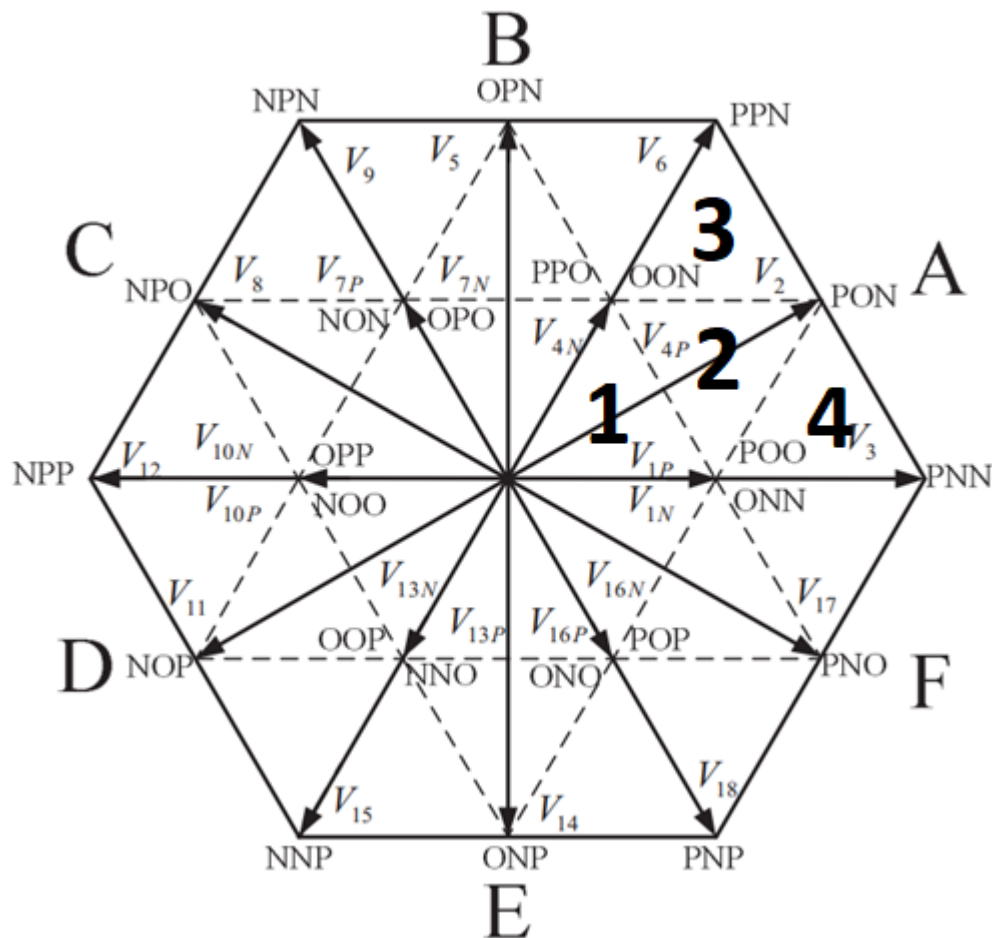


Figure 31 : Hexagone de fonctionnement SVPWM [2]

### 7.2.1 Fonctionnement

A la fin il y a deux manières d'activer les sorties et elles sont décrites aux chapitres 7.3.1 et 7.3.2.

Il y a en tout 19 vecteurs disponibles dont 7 ont des vecteurs double ou triple.

Exemple : Vecteur au centre est le vecteur 0, il est possible de l'utiliser en activant soit 'OOO', 'PPP' ou 'NNN'.

Vecteur 1 peut être utilisé avec le vecteur  $V_{1P}$  avec 'POO' ou avec le vecteur  $V_{1N}$  'ONN'.

Ces définitions correspondent aux 3 phases de l'onduleur. Si le vecteur 2 donne 'PON' il faut activer la première phase à la tension positive, la deuxième phase à tension 0V et la troisième phase à la tension négative du système.

Pour le fonctionnement il faut créer un vecteur de référence qui va se déplacer dans cette hexagone, il fait une rotation de 360 degrés chaque période. La taille du vecteur dépend de la tension de sortie voulu.

Le but est d'utiliser 3 vecteurs existant de la Figure 31 (V0 à V18) et de les additionner pour recréer le vecteur de référence. Selon la position du vecteur de référence il faut choisir différents vecteurs et pour changer la taille de ces vecteurs c'est la durée d'activation qui doit être modifiée en activant le vecteur plus ou moins longtemps avec les paramètres  $\tau_a$ ,  $\tau_b$  et  $\tau_c$ .

Le vecteur de référence est créé avec les tensions en sortie de la régulation (La régulation est la même que la méthode PMW mais les sorties ne sont pas les mêmes mais  $U_{\alpha}$  et  $U_{\beta}$ . (Figure 32). Ceci permet de créer le vecteur de référence avec une taille de  $\sqrt{U_{\alpha}^2 + U_{\beta}^2}$  et un angle de  $\text{atan2}(\frac{U_{\beta}}{U_{\alpha}})$ . Avec ces deux valeurs le vecteur de référence tournant dans l'hexagone est prêt.

## 7.2.2 Régulation pour SVPWM

La régulation dans SVPWM doit fournir une sortie  $v_{\text{ref}}$  avec la taille et l'angle.

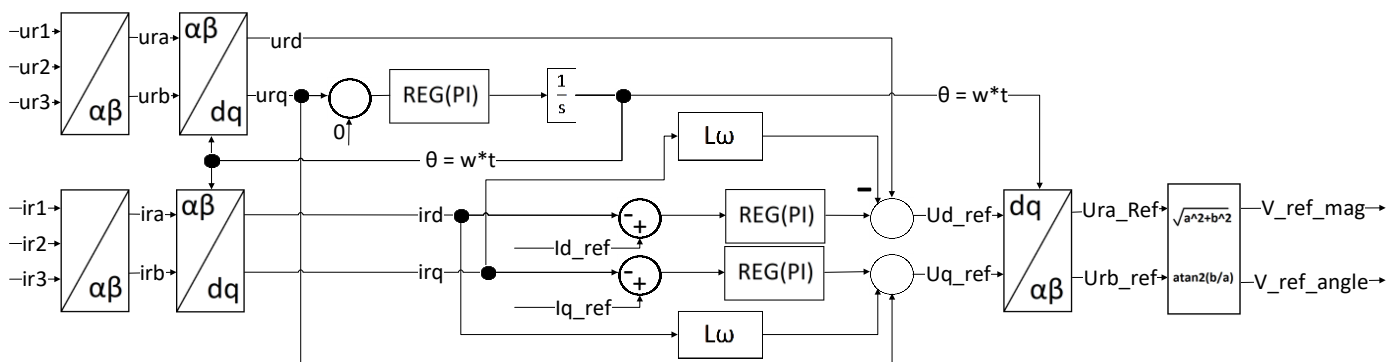


Figure 32 : Création du vecteur de référence pour SVPWM

## 7.2.3 Calcul SVPWM

Exemple avec des graphiques :

Il faut reproduire le Vecteur  $V_{\text{ref}}$  à  $25^\circ$  et 500V avec 3 vecteurs.

Selon les calculs on obtient (Formules en annexe 19.4) :

Le premier vecteur : Utilise V1 pendant 0% du temps.

Le deuxième vecteur : Utilise V2 pendant 85% du temps.

Le troisième vecteur : Utilise V3 pendant 15% du temps.

La somme de ces trois vecteurs donne un vecteur équivalent au vecteur de référence.

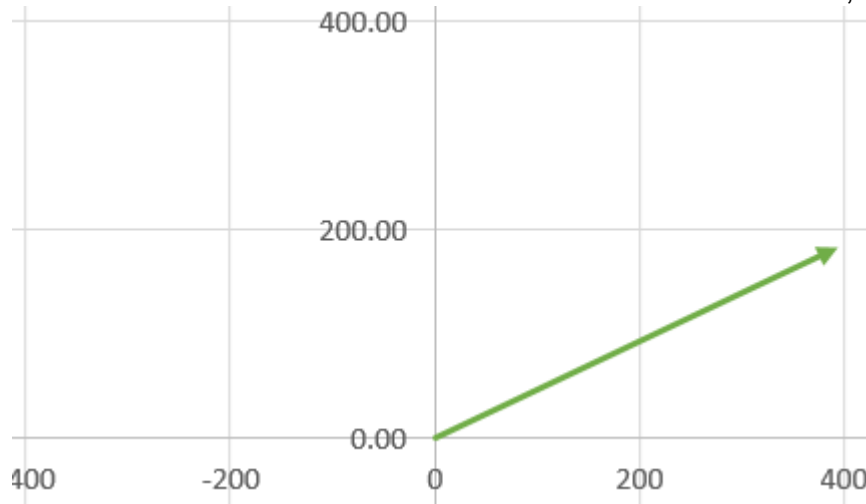


Figure 33 : Vecteur Vref

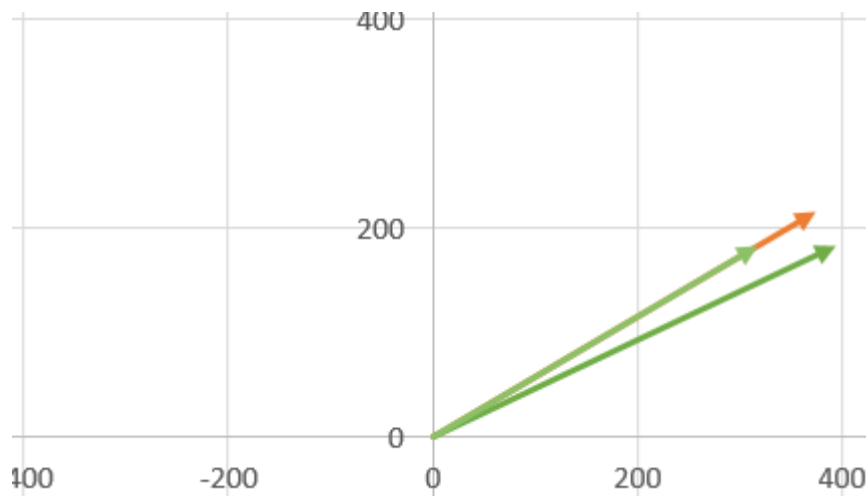


Figure 34 : On utilise le vecteur orange V2 pendant 85% du temps

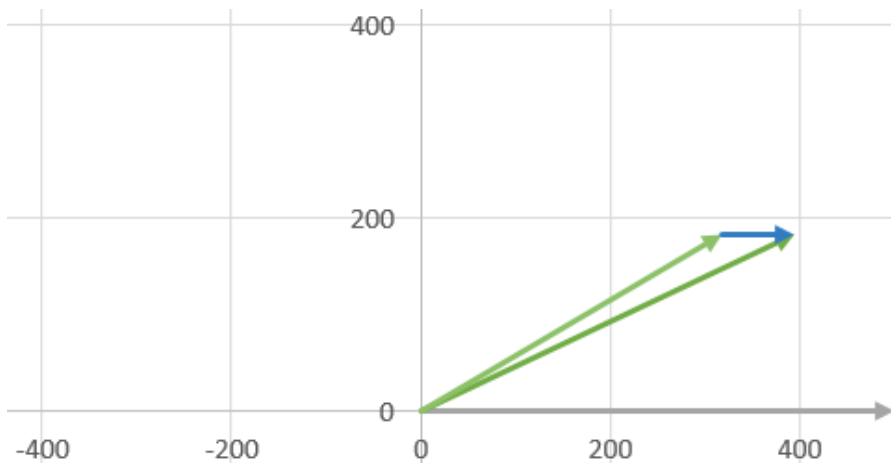


Figure 35 : On utilise le vecteur gris V3 pendant 15% du temps et additionne ces vecteurs

#### 7.2.4 Comment choisir les vecteurs à utiliser pour reproduire le $v_{ref}$ ?

Pour trouver les vecteurs à choisir il y a plusieurs calculs à faire :

- Dans quel secteur est le vecteur ? (A à F ou 1 à 6), Figure 163
- Dans quelle région est le vecteur ? (1 à 4 dans chaque secteur), Figure 164
- Temps d'activation des vecteurs. ( $\tau_a$ ,  $\tau_b$  et  $\tau_c$ ), Figure 165

Les formules pour ces calculs sont en annexe (19.4).

Une fois que tous ces calculs sont faits il est possible de sélectionner les bons vecteurs avec le secteur et la région puis le temps d'activation de ces vecteurs.

Selon la région on change les vecteurs.

Sélectionne toujours 3 vecteurs et dans la **région 1** il y a toujours le vecteur 0 dans tous les cas.

On multiplie le premier vecteur par tau\_a, le deuxième avec tau\_b et le dernier par tau\_c pour obtenir la bonne somme de vecteur.

Les vecteurs pour le secteur 1 sont les suivants →

### Vecteur à choisir Secteur 1

Multiplie par →	Tau_a	Tau_b	Tau_c
Région 1	V1	V0	V4
Région 2	V1	V2	V4
Région 3	V6	V2	V4
Région 4	V1	V2	V3

Tableau 4 : Tableau des vecteurs à choisir selon la région pour le secteur 1

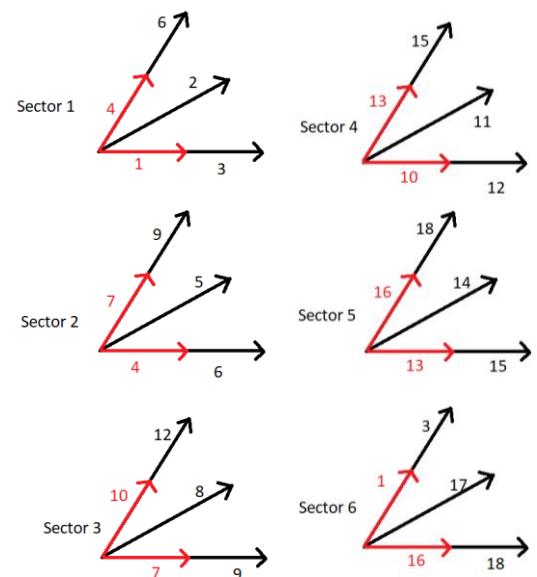


Figure 36 : Vecteurs dans les secteurs de 1 à 6. Tous dans la même orientation

Pour choisir les vecteurs dans les secteurs suivants il suffit de prendre les vecteurs aux mêmes positions que pour le premier secteur. Avec la Figure 36

Ou avec la formule suivante :

(Vecteur correspondant dans S1) + 3\*(secteur-1)

Exemple : Pour le secteur quel est le long vecteur en diagonale, qui correspond au vecteur 2 dans le secteur 1 →  $(2) + 3 \cdot (4 - 1) = \text{vecteur } 11$  à choisir

**Attention ! : Si la formule donne un vecteur plus grand que 18, il faut soustraire 18.**

Selon le secteur et selon la région il faut sélectionner des vecteurs différents.

Exemple pour le secteur 4 les vecteurs suivants sont choisis de la même manière : Suivre les données de la Figure 36 ou utiliser la formule au-dessus (la formule est importante lorsqu'on fait le choix avec du code).

**Vecteur à choisir Secteur 4**

Multiplie par →	Tau_a	Tau_b	Tau_c
Région 1	V10	V0	V13
Région 2	V10	V11	V13
Région 3	V15	V11	V13
Région 4	V10	V11	V12

*Tableau 5 : Tableau des vecteurs à choisir selon la région pour le secteur 4*

En annexe, il y a les vecteurs pour tous les secteurs et régions. (19.5)

**7.3 Outputs avec SVPWM**

Une fois que les calculs sont faits, il est nécessaire de calculer comment activer la state machine des cartes de puissance. Il y a deux méthodes.

SVPWM\_1 qui est une version simple et SVPWM\_2 qui est la version complexe. Le fonctionnement est le même pour les deux cas mais la manière d'activer les sorties sont différentes pour SVPWM\_1 et SVPWM\_2. La première est une addition des composantes et la deuxième est une séquence.

**7.3.1 Activation SVPWM\_1, version simple**

Exemple d'activation sur l'installation :

Avec les valeurs de l'exemple des graphiques aux figures suivantes Figure 33 à Figure 35 les résultats pour les activations sont les suivantes.

Avec les vecteurs v1, v2 et v3 qui correspondent aux points POO, PON et PNN. Il faut les séparer par phase puis les additionner.

Vecteur	Temps	Phase 1	Phase 2	Phase 3
V1 POO	0%	0% P	0% O	0% O
V2 PON	85%	85% P	85% O	85% N
V3 PNN	15%	15% P	15% N	15% N
Somme →	<b>100%</b>	<b>100% P</b>	<b>15 % N 85% O</b>	<b>100% N</b>

*Tableau 6 : Tableau des valeurs de modulation selon les choix des vecteurs*

En appliquant les valeurs obtenues en paramètre aux modulateurs des phases →

Phase 1 : Mettre le modulateur tout le temps à la tension positive (100% P)

Phase 2 : Mettre le modulateur à 0.15 sur la tension négative. (85% du temps il se met à 0V)

Phase 3 : Mettre le modulateur tout le temps à la tension négative (100%N)

Ce cas précis fonctionne uniquement pendant une très courte durée et au cycle de calcul suivant les nouvelles valeurs sont appliqués aux modulateurs.

**7.3.2 Activation SVPWM\_2, version complexe :**

**ATTENTION** : Cette méthode nécessite de faire une transformation complète du programme Poetic en changeant totalement le fonctionnement de base : Lire avant la partie comparaison des deux méthodes. Au chapitre 7.4 !

Après la comparaison sur Matlab des deux méthodes SVPWM\_1 et SVPWM\_2 il est possible d'utiliser la plus simple des deux, SVPWM\_1.

La méthode SVPWM\_2 n'est pas utilisée.

Mais pour faire la comparaison sur Matlab il faut comprendre son fonctionnement.

### 7.3.3 Fonctionnement SVPWM\_2

Jusqu'au calculs c'est exactement la même méthode. Mais l'activation est une séquence spéciale et pas des activations pendant une certaine durée. Dans ce cas il faut suivre une séquence entre chaque cycle de calcul.

Une fois les valeurs calculées les semi-conducteurs s'activent pendant  $\tau_a / 2$  avec le vecteur voulu, puis pendant le temps restant de  $\tau_a / 2$  le vecteur avec la version alternative et la même chose pour  $\tau_b$  et  $\tau_c$ .

- Le  $\tau_b$  s'active deux fois  $1/2$  du temps à 'PON'
- Le  $\tau_c$  s'active deux fois  $1/4$  du temps à 'PPO' et deux fois  $1/4$  du temps à 'OON'
- Le  $\tau_a$  s'active deux fois  $1/4$  du temps à 'POO' et une fois  $1/2$  du temps à 'ONN'

L'ordre choisi pour ces activations permet d'éviter un déséquilibre, mais aussi assure de faire qu'un seul changement d'un état à l'autre. Dans cette méthode le  $\tau_a$ ,  $\tau_b$  et  $\tau_c$  ne sont pas lié au vecteur directement, mais plutôt les timings pour activer les séquences suivantes.

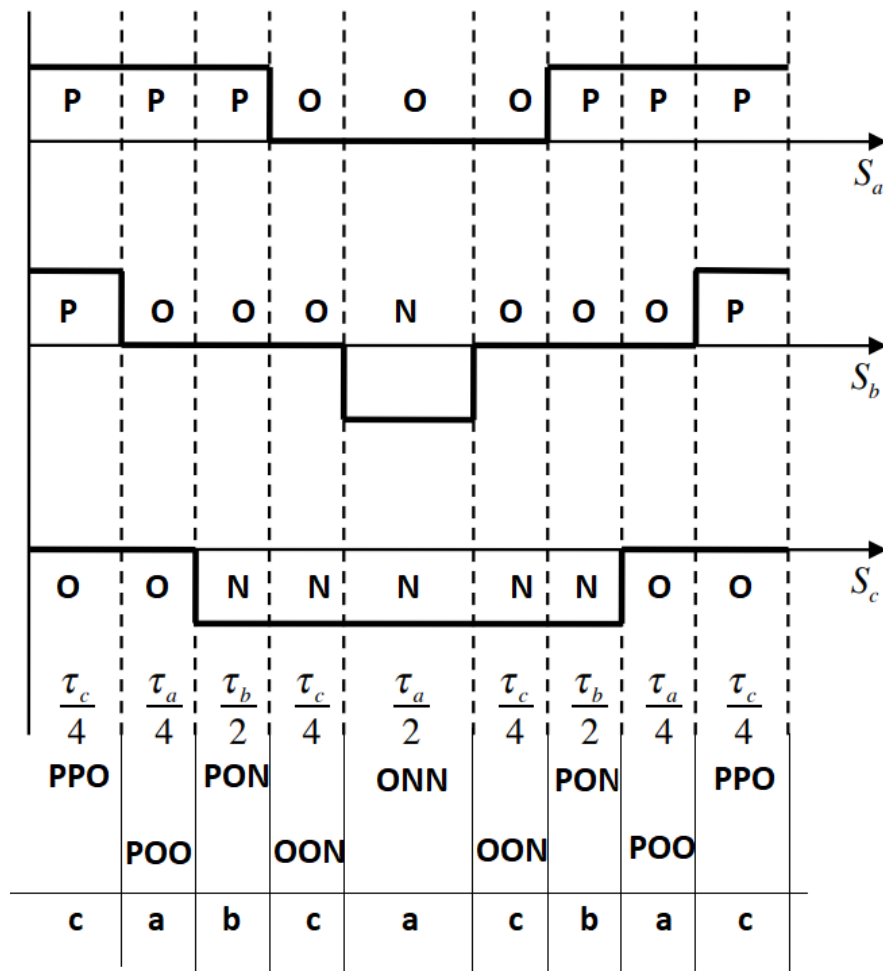


Figure 37 : Séquence d'activation pour la méthode complexe SVPWM\_2

#### 7.4 Comparaison des méthodes SVPWM\_1 et SVPWM\_2

La méthode SVPWM\_2 nécessite une transformation complète du code Poetic il faut être sûr que les avantages sont assez bons pour justifier cette modification. C'est pour cette raison qu'une comparaison est faite sur Matlab avec les deux fonctionnements.

Résultats avec explications et les images à la Figure 38 :

Le temps nécessaire pour faire ces modifications ne semble pas être utile car les sorties du système obtenu ne varient que très peu. Exemple avec ces paramètres : Référence à 350V avec un Bus DC à 750VDC.

La sortie de la phase 1 est différente uniquement au changement de la partie positive à négative sur la version bleu SVPWM\_2. Elle passe par les trois états en même temps. Mais aussi une partie du temps le système est saturé à 1 dans la version SVPWM\_1. Cette saturation est dû au fait que dans le secteur 1 la phase 1 est uniquement positive car les vecteurs alternatifs ne sont pas utilisés.

**Mais** une correction est possible dans le code. Car consommer uniquement de la tension positive veut dire que le condensateur avec la tension DC supérieure va se décharger plus rapidement. Si on contrôle ce changement de tensions DC sur les condensateurs il est possible de commuter pendant cette période et de ne pas rester à 1 tout le temps. Il n'est donc pas utile d'utiliser la méthode SVPWM\_2 dans notre cas. Dans le futur pour améliorer le système et le rendre

plus équilibré la méthode SVPWM\_2 peut être utilisé. Comparaison à la Figure 38.

Les graphes correspondent aux modulations à mettre sur les phases. (Sortie P, N et O du chapitre 5) Orange = SVPWM\_1, Bleu = SVPWM\_2

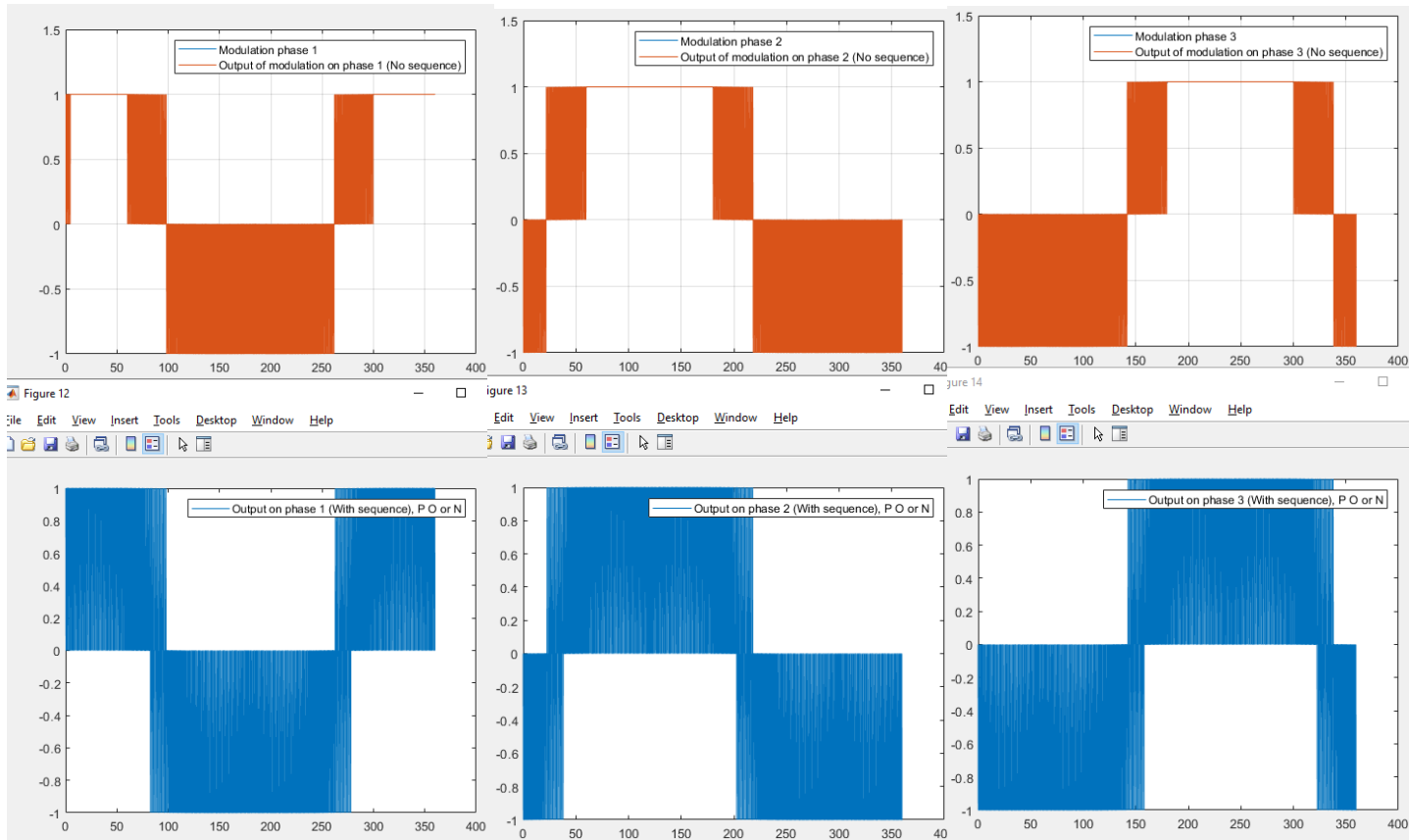


Figure 38 : Comparaison entre les méthodes SVPWM\_1 et SVPWM\_2 pour les sorties phase 1,2 et 3. Disponible en annexe

### 7.5 Fichier Excel pour les tests SVPWM au départ

Il y a un fichier Matlab qui fait les calculs très précisément, mais pour comprendre le fonctionnement de base il y a eu un début de développement sur un fichier Excel avec 1 point chaque degré (360 points) pour comprendre les différents calculs et comment il allait faire varier le système et réagir avec les autres paramètres.

Une fois que le tableau est en ordre et les graphiques concluant, il est nécessaire d'être plus précis et avec Matlab c'est possible. Avec Matlab la précision peut facilement dépasser le millième de degré par calcul sans ajouter un nombre énorme de cellules. Simplement en changeant un seul paramètre dans le script Matlab.





## 8.2 ANPC Triphasé

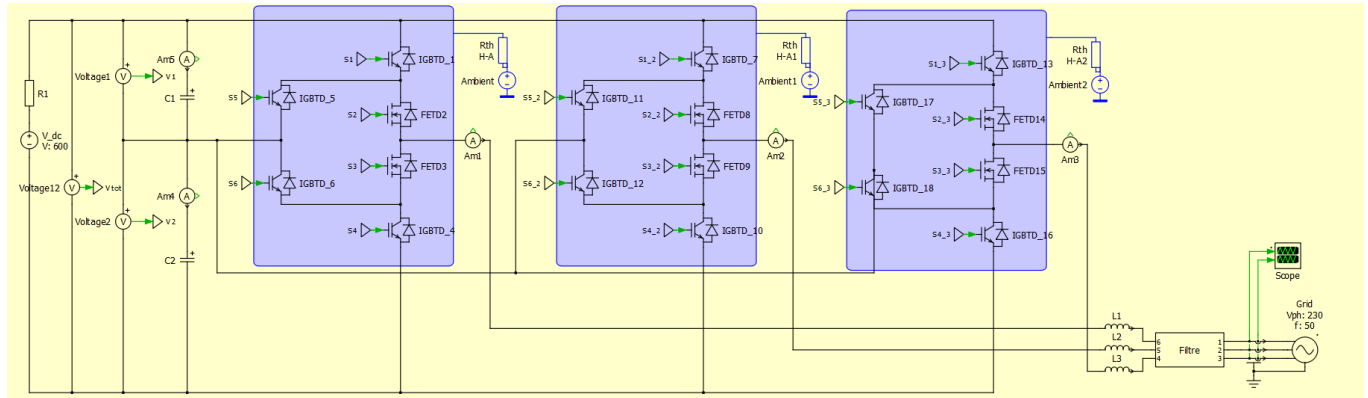


Figure 41 : Schéma Onduleur ANPC Triphasé avec filtre

Le modèle ANPC contient 3 bras d'onduleur avec 6 semi-conducteurs (IGBT, Mosfet ou Hybride) par phase. La sortie de phase passe par un filtre et est connecté sur le réseau 325V/50Hz. Chaque phase possède sa machine d'état pour piloter ses semi-conducteurs. Pour contrôler les courants qui sont injectés sur le réseau il y a une régulation qui pilote les 3 phases.

## 8.3 Régulation

La partie régulation du système permet de choisir un courant à injecter dans le réseau ou de choisir les puissances actives et réactives.

Sur ce schéma il y a les mesures de courants et tensions avec les courants de référence à choisir. En sortie il y a les tensions de références pour la modulation. Ces tensions de référence doivent encore être normalisé entre +1 et -1 et aussi transformé en signaux PWM pour que la state machine sur chaque phase puisse lire les valeurs.

Répétition du chapitre régulation pour aider à la simulation PLECS (Chapitre 7.1)

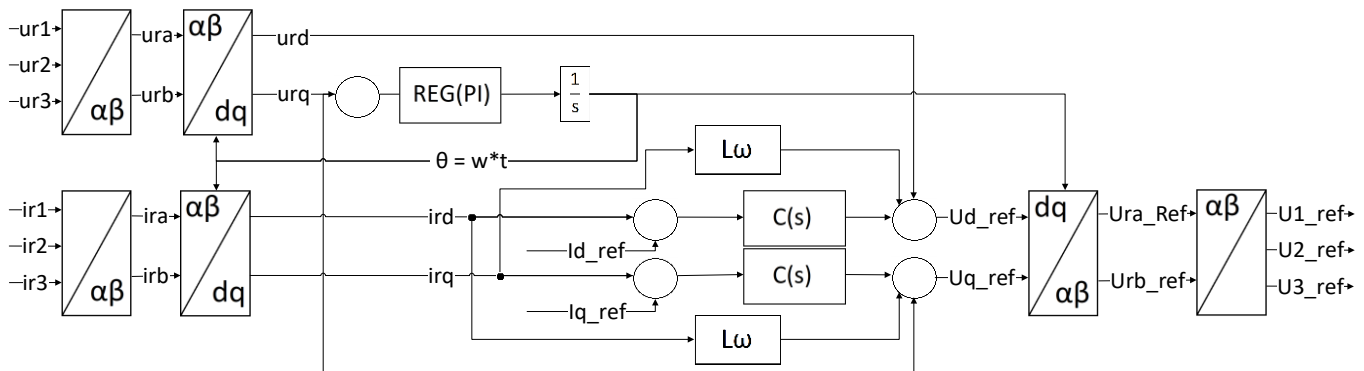


Figure 42 : Schéma bloc de régulation (C'est la même que la Figure 30)

Sur le schéma de la simulation Plecs la partie de normalisation du signal et transformation du signal vers des signaux PWM est disponible à la Figure 43.

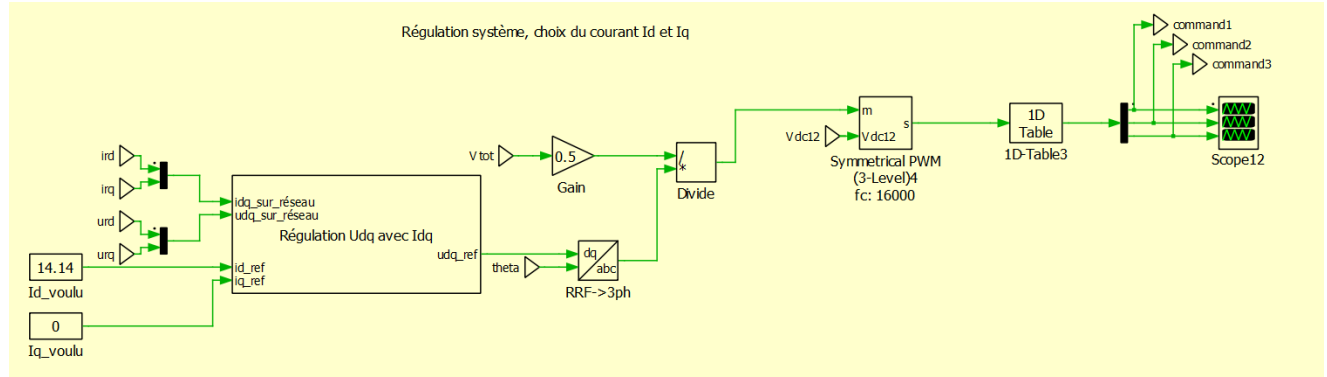


Figure 43 : Schéma avec les consignes de courants et conversions des signaux pour la machine d'état

La partie suivante est à l'intérieur du bloc « Régulation Udq avec Idq » de la Figure 43.

Il y a une comparaison entre les courants de réseau et les courants de référence pour déterminer la tension de référence udq nécessaire.

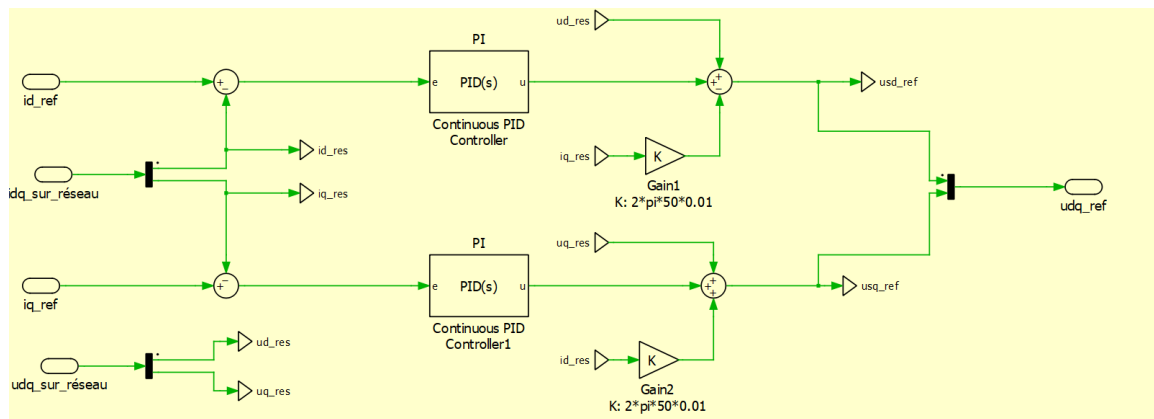


Figure 44 : Schéma des régulateurs PI pour la tension de référence des PWMs avec les courants id et iq

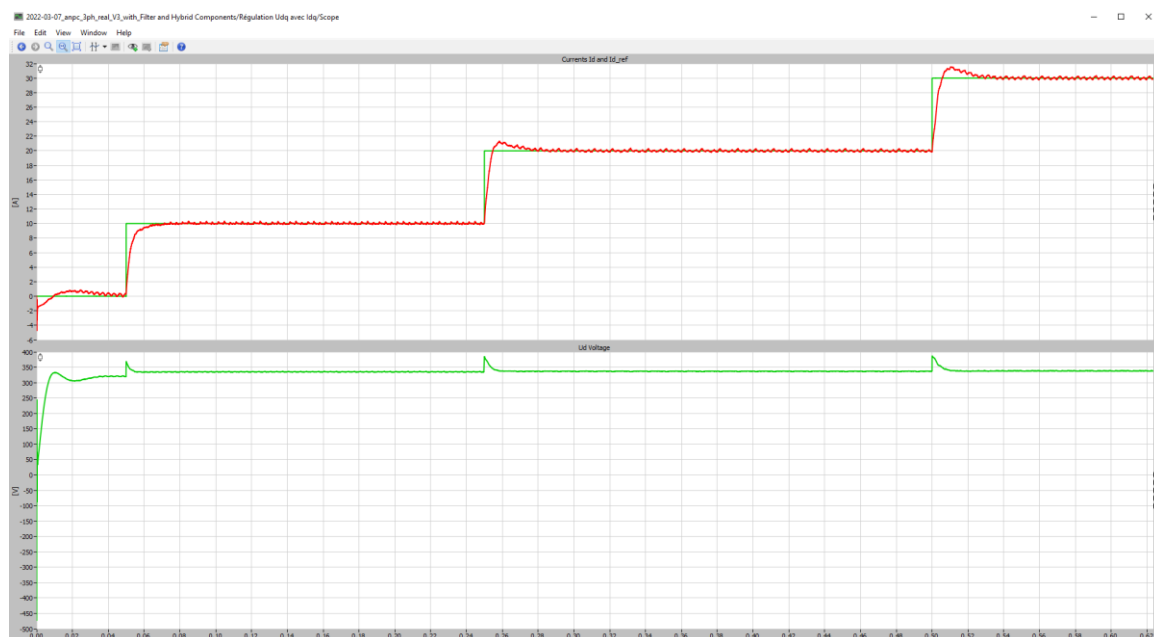


Figure 45 : Régulation sur le courant Id (Même chose se passe sur Iq); Graphique 1 : Courant voulu par l'utilisateur et le courant fourni, Graphique 2 : Tension en sortie du régulateur Ud

Avec la tension de référence  $u_{dq}$  sortante de la Figure 44 il faut la transformer en signaux pour la machine d'état. Premièrement en passant du système biphasé en dq à triphasé et de normaliser la tension avec la tension du bus DC.

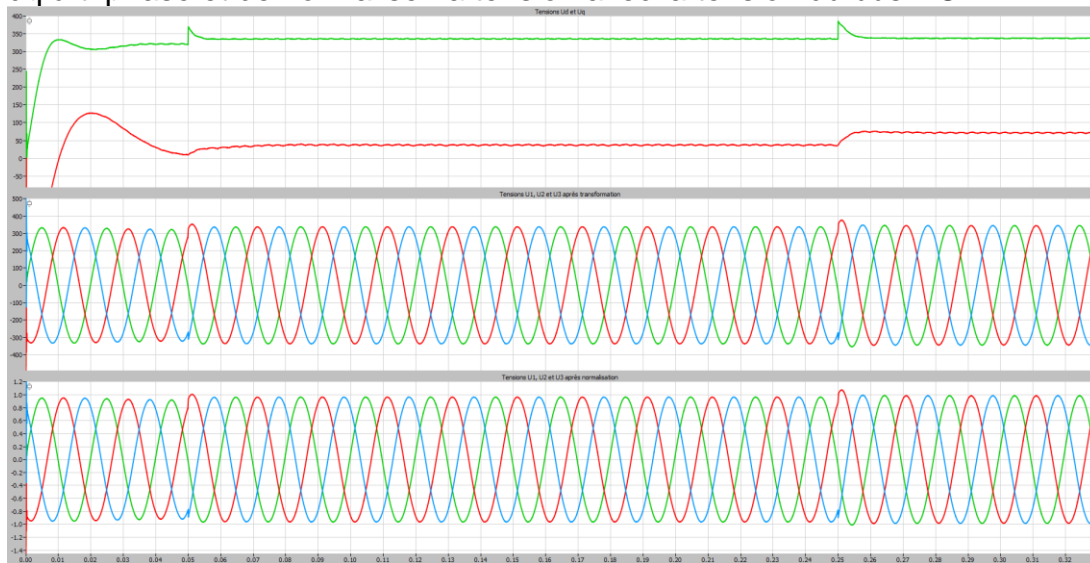


Figure 46 : 1ère courbe : Tensions  $U_d$  et  $U_q$ , 2ème courbe : Les tensions de référence en sortie des régulateurs, 3ème : Tensions normalisées (entre -1 et 1) Faible changement entre les changements de consigne

Sortie du système avec les courants et les puissances ainsi que la tension de réseau et la tension sur le bus DC

(Disponible en annexe)

Consigne de courants  $I_d$  changeante :

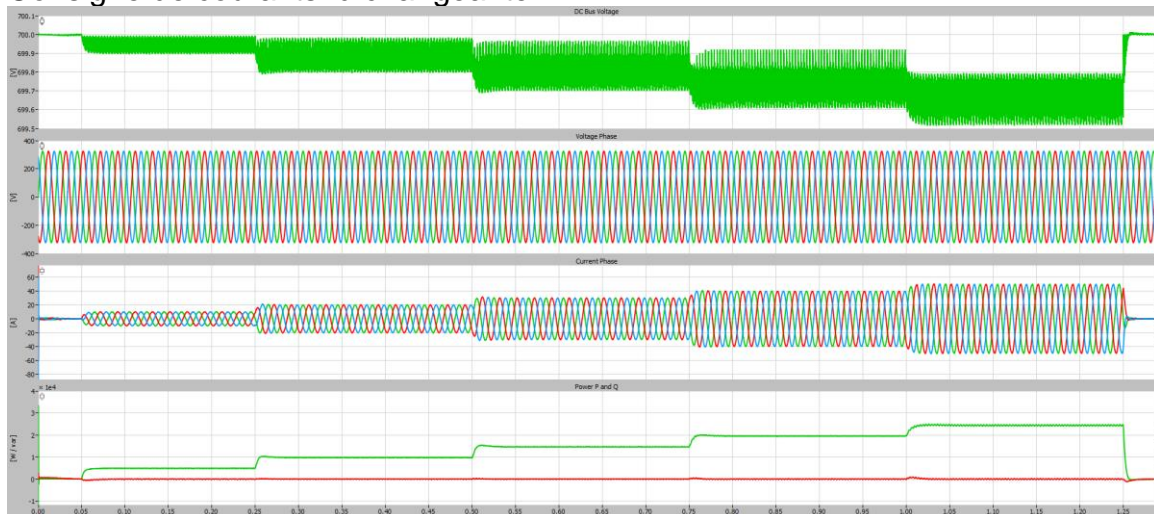


Figure 47 : Graphique 1 = Courbe de tensions bus DC, Graphique 2 = Tensions de réseau, Graphique 3 = Courants de réseau, Graphique 4 = Puissance actives et réactives (Disponible en annexe)

### 8.4 Résultat protection de la state machine

Lorsqu'un problème se produit la state machine des cartes de puissances change d'état et ne suit pas la procédure habituelle et coupe l'installation en suivant les états définis au chapitre 5.3.

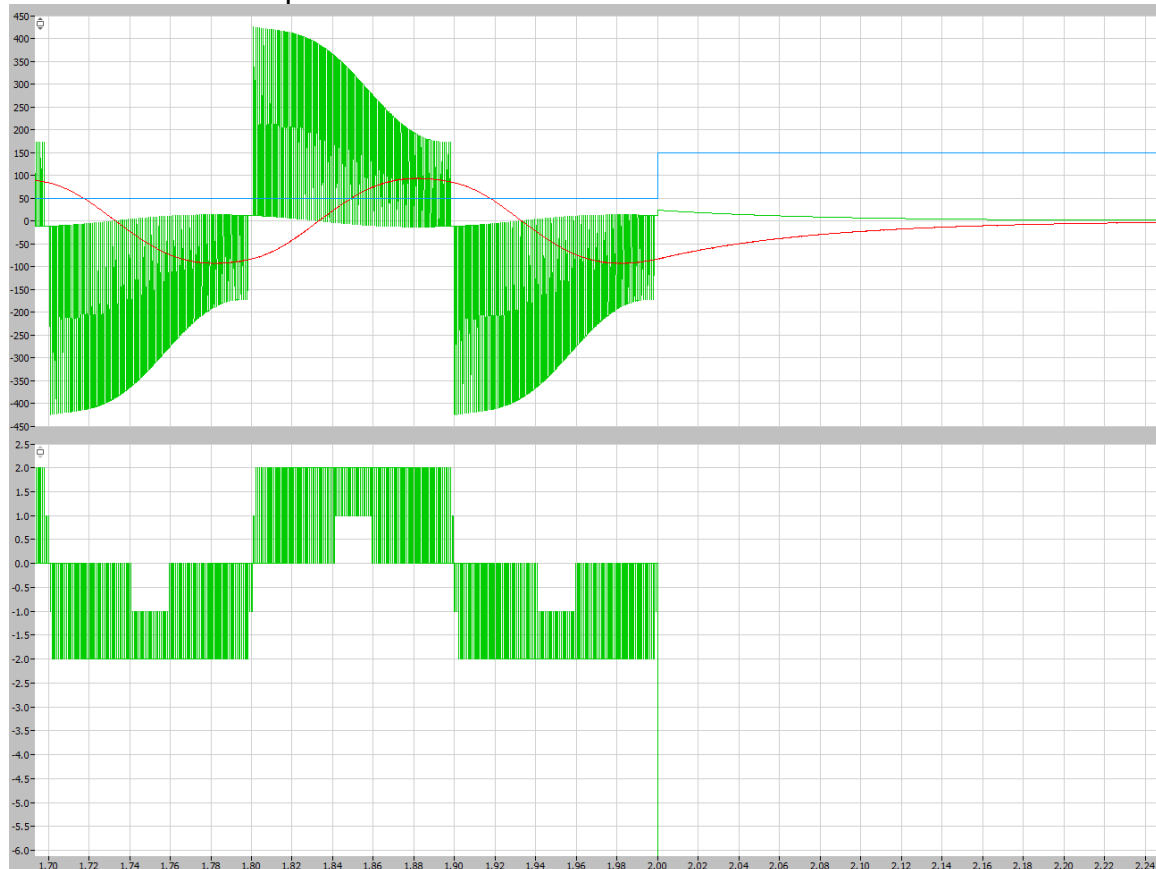


Figure 48 : Graphique 1 = Courants et tensions en sortie du système lorsqu'il y a un défaut sur un semi-conducteur, avec en bleu le défaut détecté. Graphique 2 = Etat de la state machine et si elle se trouve entre -2 et 2 c'est le fonctionnement normal

## 9 Détermination de pertes et limitations en fonction des algorithmes et des configurations (Si, SiC, Hybride)

Pour déterminer les pertes dans le système il y a une simulation avec les données du fabricant pour les semi-conducteurs (Mosfet et IGBT). Les deux types de pertes à contrôler sont les pertes de conductions et les pertes de commutation. Ces pertes dépendent de la température des composants, il faut donc déterminer les informations thermiques tel que le dissipateur thermique. Dans notre cas nous avons un dissipateur de la marque fischer elektronik qui se connecte sur les 6 semi-conducteurs d'une phase. Ce qui donne 3 dissipateurs pour un système triphasé. Avec une conduction thermique :

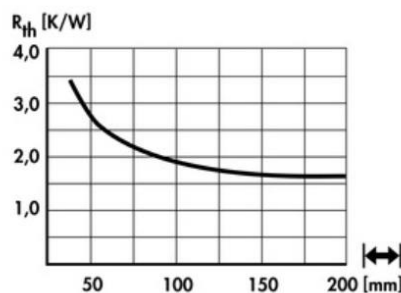


Figure 49 : Dissipateur thermique Fischer Elektronik (SK 508 1000 SA) de 180 mm de long

Notre modèle est coupé à 180mm et possède donc une conduction thermique de 1.7K/W. Dans la Figure 50 le grand bloc bleu dans la simulation correspond à ce dissipateur thermique on y ajoute la résistance thermique trouvé avec une température ambiante.

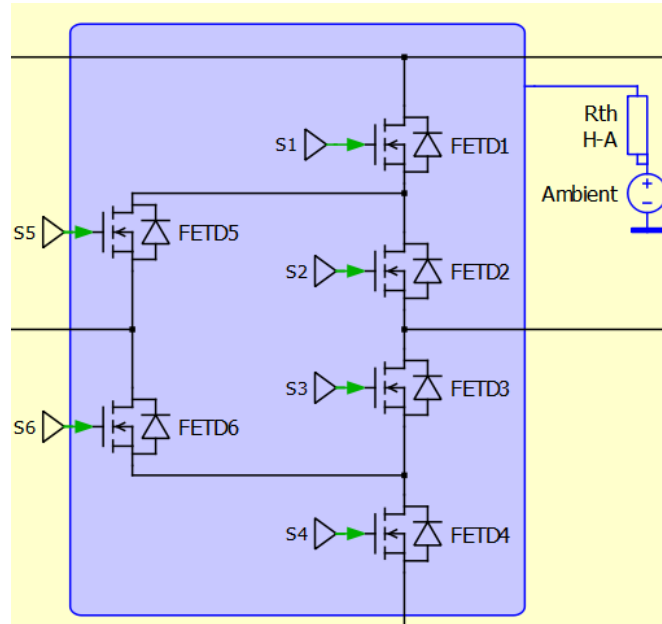


Figure 50 : Modélisation thermique dans la simulation

## 9.1 Résultats Pertes

Pour les comparaisons des pertes suivantes toutes les mesures sont faites avec des state machine sur les cartes de puissance PWM4. La comparaison est entre des modèles avec uniquement des IGBTs, uniquement des Mosfets et une combinaison hybride. Pour chaque modèle ces tests se font à plusieurs niveaux de courants. L'installation est dimensionnée pour un courant max de 50 A.

Les pertes sont en watt. Les pertes totales sont les pertes de commutations et les pertes de conductions pour tous les semi-conducteurs sur les trois phases. Ceci correspond à deux types de pertes sur 18 semi-conducteurs pour le système complet.

### 9.1.1 Mosfet

Id [A]	Pertes tot [W]	Puissance Source [W]	Rendement [%]
10.0	102.284	5156.6	97.9382
20.0	127.041	10039	98.7394
30.0	152.99	15299.3	99.0052
40.0	189.879	20997.9	99.0879
50.0	238.391	26497.8	99.0982

Tableau 7 : Résultats des pertes et rendements pour le Mosfet

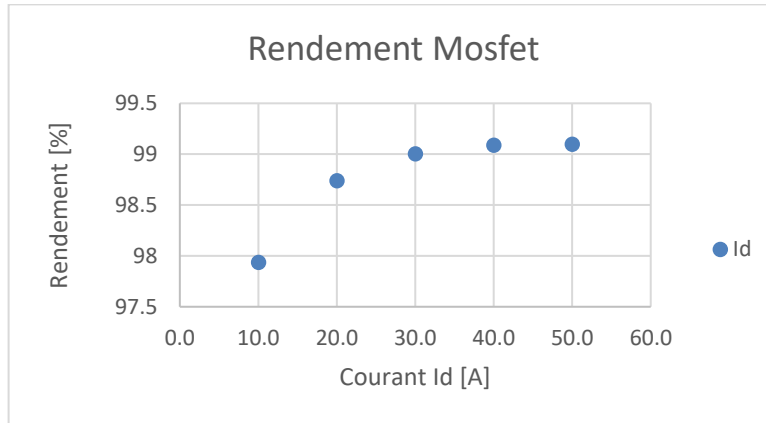


Figure 51 : Rendement Système avec Mosfet

### 9.1.2 IGBT

Id [A]	Pertes tot [W]	Puissance Source [W]	Rendement [%]
10.0	156.074	4817.94	96.8022
20.0	322.598	9612.98	96.6985
30.0	481.905	14997	96.7112
40.0	571.382	19580.5	97.0818
50.0	663.118	2420.4	97.3002

Tableau 8 : Résultats des pertes et rendements pour l'IGBT

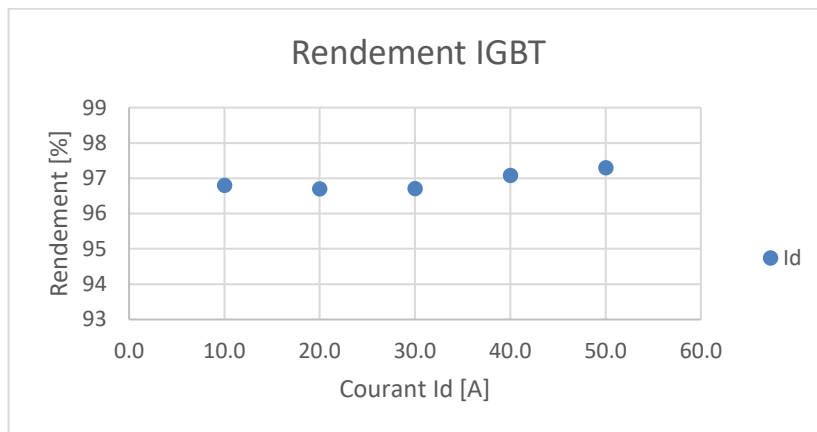


Figure 52 : Rendement Système avec IGBT

### 9.1.3 Hybride

Topologie Hybride = S2 et S3 Mosfet et les autres IGBT

Id [A]	Pertes tot [W]	Puissance Source [W]	Rendement [%]
10.0	146.106	4933.98	97.0027
20.0	276.418	10184.7	97.1887
30.0	365.675	15249.9	97.5425
40.0	439.437	20389.8	97.8123
50.0	508.9	25162.9	97.987

Tableau 9 : Résultats des pertes et rendements pour la topologie Hybride

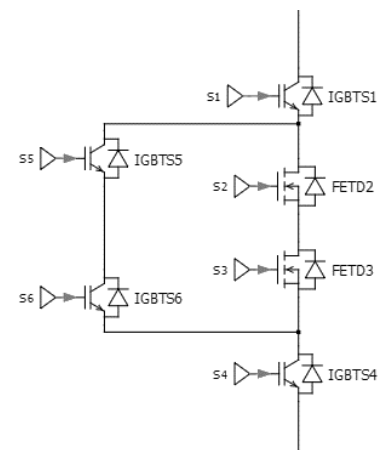


Figure 53 : Topologie Hybride



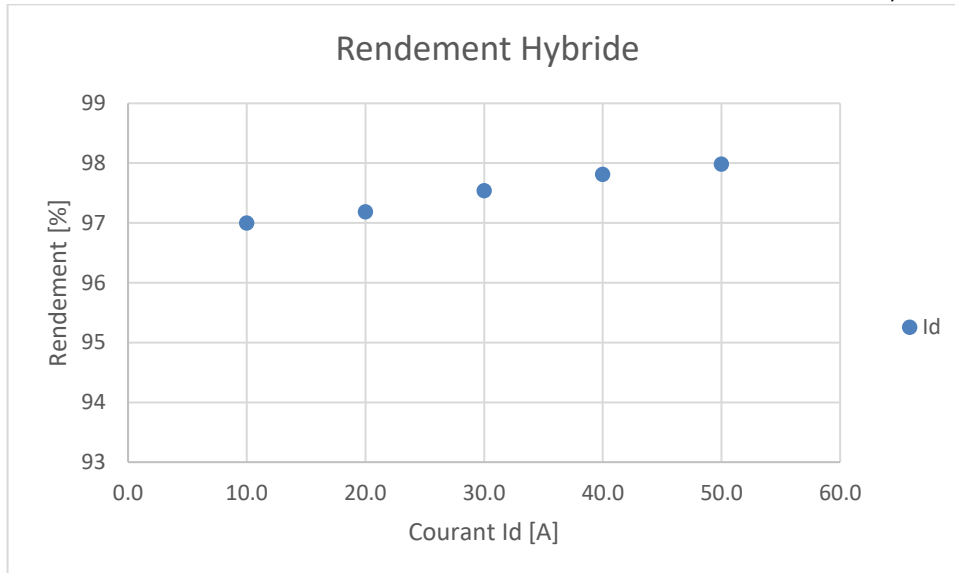


Figure 54 : Rendement Système en Hybride

## 9.2 Conclusion

En comparant les trois méthodes :

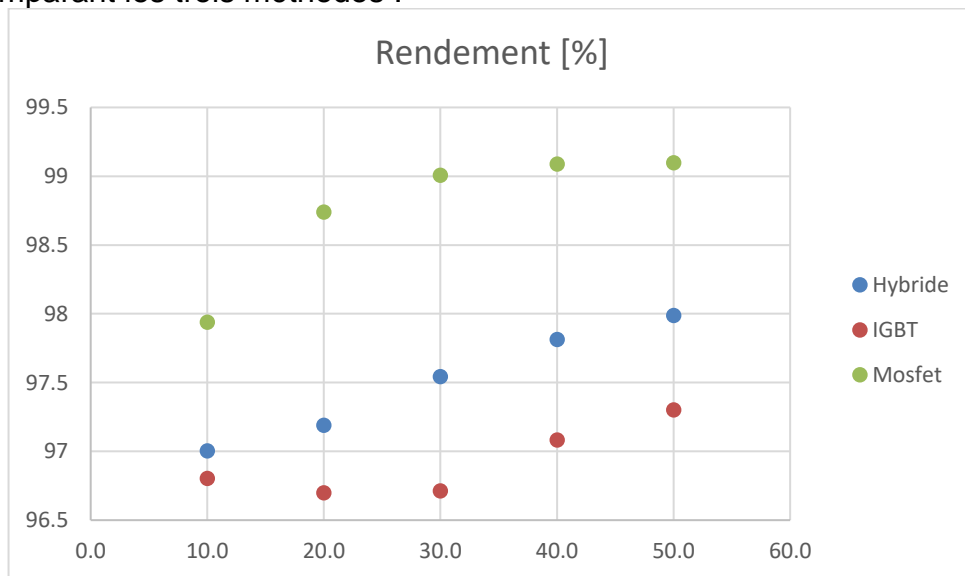


Figure 55 : Rendements pour les trois méthodes de semi-conducteurs

Prix de la carte de puissance = 190.04 CHF

Module Mosfet = 127.35 CHF (2 Mosfet par module, donc 3 modules nécessaires)

Module IGBT = 61.49 CHF (2 IGBT par module, donc 3 modules nécessaires)

Carte + 3 modules Mosfet = 572.09 CHF pour une seule carte de puissance

Carte + 3 modules IGBT = 374.51 CHF pour une seule carte de puissance

Donc par carte le prix avec des Mosfet est 1.5 fois plus grand qu'une carte IGBT.

Le semi-conducteur avec le plus de pertes est l'IGBT. Le Mosfet a de très peu de pertes et permet d'avoir un très bon rendement. Pour la méthode Hybride les pertes sont entre les deux.

Cependant le prix des Mosfets est beaucoup plus important. Pour par exemple 50 cartes de puissance la différence de prix est de 10'000 CHF.

Si le système est uniquement des IGBT il peut également avoir des limites de fonctionnement à ne pas négliger. La vitesse de commutation du Mosfet est de 100MHz pour un courant de 50A et encore plus rapide à plus faible courant. Celle de l'IGBT est de 10kHz pour un courant de 50A. Le système ANPC fonctionne à 20kHz et s'il faut aller plus vite ou augmenter le courant sur l'IGBT ce n'est peut-être pas une option. L'IGBT peut fonctionner à 20kHz uniquement pour un courant de 30A.

Il faut donc faire un choix avec le nombre d'installations qui sont envisagés et le rendement pour choisir lequel est mieux adapté à la situation.

## 10 Programmation POETIC CPU et méthodes de modulation

La programmation Poetic correspond au cerveau du système. Poetic est responsable de la gestion de la state machine du système (Chapitre 10.1), mais aussi la régulation de l'installation.

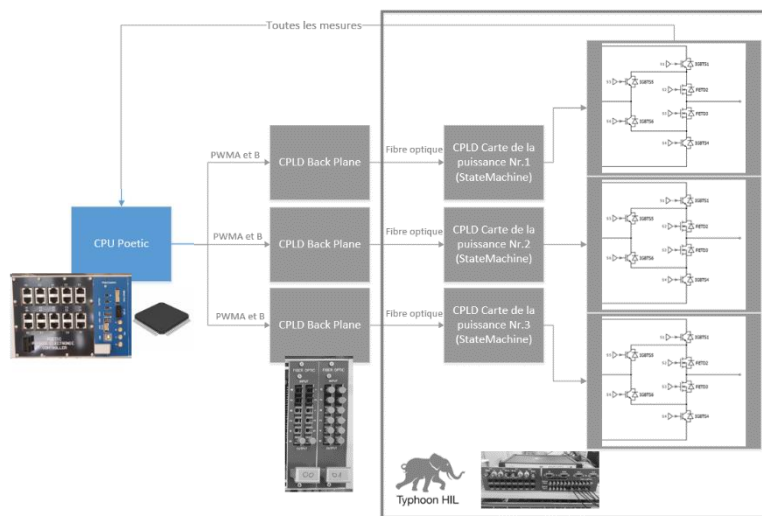


Figure 56 : Ce chapitre traite de la programmation du CPU de Poetic ou la partie qui est en couleur sur cette image

### 10.1 State Machine du Système

La state machine du système n'est pas la même que la state machine sur les cartes de puissance. Les cartes de puissance permettent de choisir comment activer les semi-conducteurs. Mais la state machine du système permet d'allumer ou d'éteindre le fonctionnement. Il y a plusieurs étapes pour activer l'onduleur ANPC. Il faut appuyer sur des boutons dans le bon ordre et des contrôles sont font (Par exemple : si la tensions DC n'est pas assez élevée, le système ne s'allume même pas).

Mais surtout un contrôle d'erreur qui coupe l'installation en mettant la machine en erreur pour un grand type de cas différents. Cette state machine du système est disponible à la Figure 57 ou en annexe 19.8.

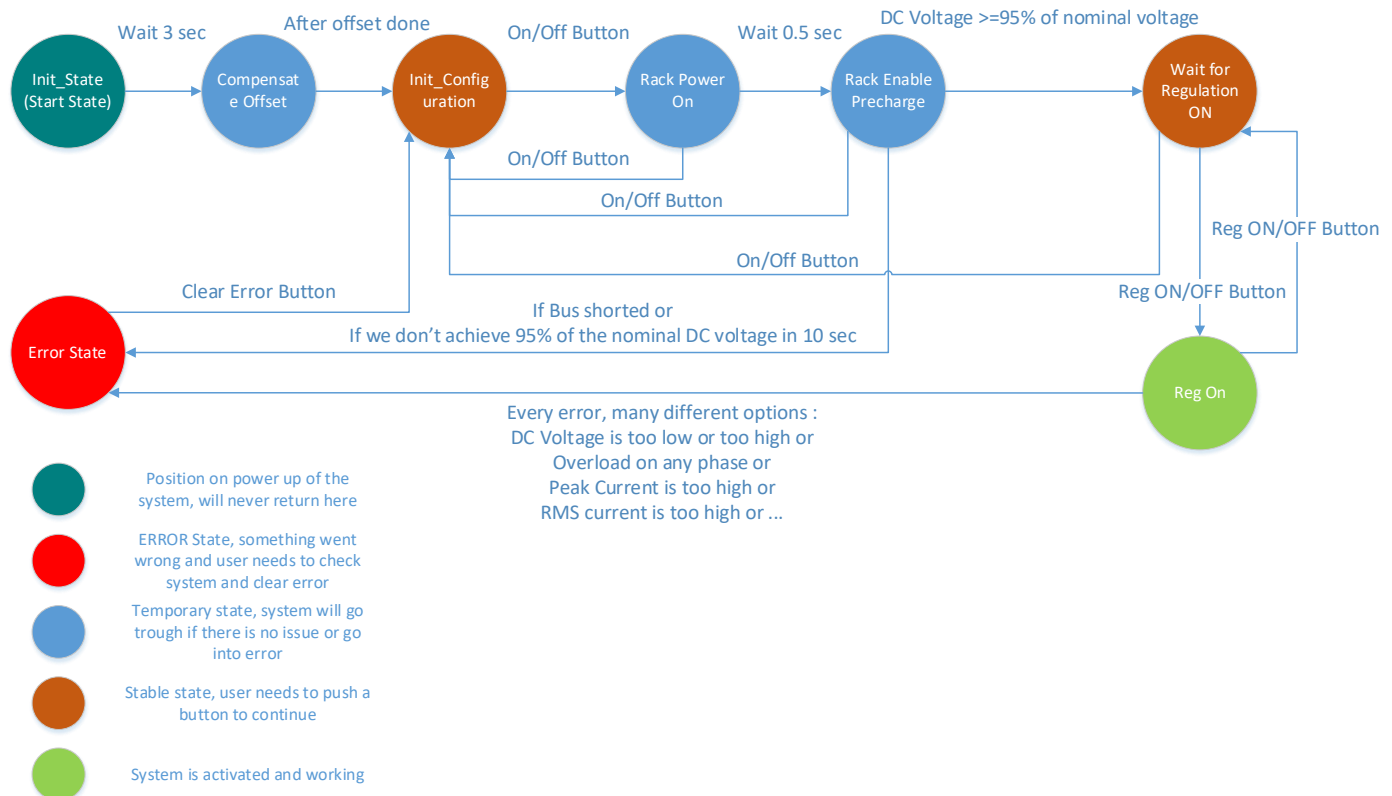


Figure 57 : Machine d'état du système

## 10.2 Programme CPU

Le programme commence dans le main.c et s'occupe de faire toutes les initialisations nécessaires, ensuite le programme passe dans une boucle infini avec quelques fonctions.

Mais le fonctionnement du programme se trouve principalement dans l'interrupt. L'interrupt s'active une fois que le CPU a fini de convertir toutes les valeurs de mesures. Une fois que l'interrupt commence il y a beaucoup d'étapes à faire (En rouge à la Figure 58). Quelques exemples : Contrôle des alarmes ou la régulation du système qui se trouvent dans process\_control\_fast (Figure 60).

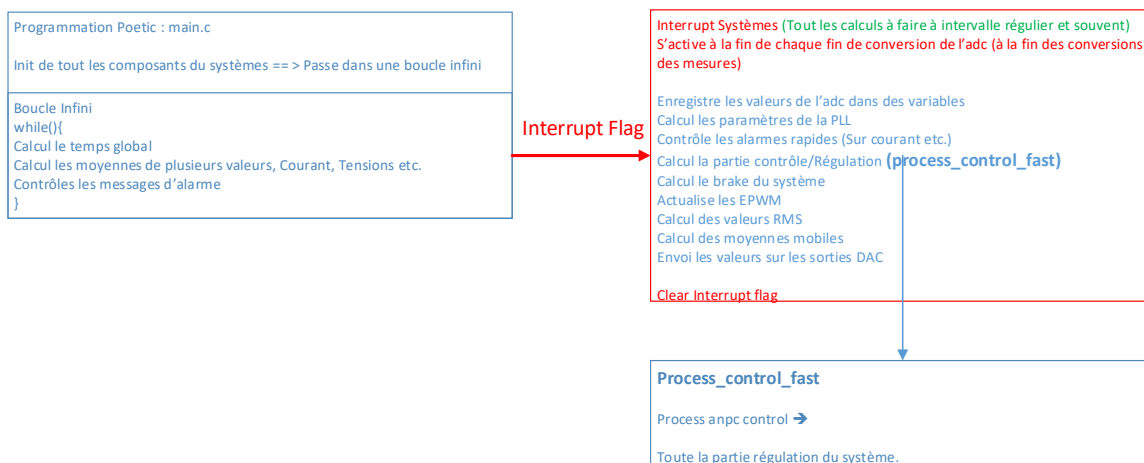


Figure 58 : Fonctionnement du code (Disponible en annexe)

## 10.2.1 Code important sur le CPU Poetic

Il y a plusieurs initialisations à faire mais la plus importante est celle des EPMW : Il faut bien configurer les registres pour que les sorties fibres optiques fournissent les signaux voulus.

Les lignes importantes à comprendre sont les lignes 119, 143-146 (A modifier dans le programme). Les autres lignes sont semblables avec d'autre projets et ne changent pas.

```

114 void init_epwm1(void){
115
116     EPwm1Regs.TBCTL.bit.FREE_SOFT = 0x02;           // free run
117
118     // Setup TBCLK
119     EPwm1Regs.TBPRD                = PWR_IN_CNT_PRD; // Set timer period 2500
120     EPwm1Regs.TBPHS.all             = 0;
121     EPwm1Regs.TBPHS.bit.TBPHS      = 0x0000;
122     EPwm1Regs.TBCTR                 = 0x0000;         // Clear counter
123
124     // Set Compare values
125     EPwm1Regs.CMPA.bit.CMPA        = 0;             // Set compare A
126     EPwm1Regs.CMPB.bit.CMPB        = 0;             // Set Compare B
127
128     // Setup counter mode
129     EPwm1Regs.TBCTL.bit.CTRMODE     = TB_COUNT_UPDOWN; // Symmetrical mode
130     EPwm1Regs.TBCTL.bit.PHSEN       = TB_DISABLE;     // Master module enable phase loading
131     EPwm1Regs.TBCTL.bit.PRDL        = TB_SHADOW;
132     EPwm1Regs.TBCTL.bit.SYNCSEL     = TB_SYNC_DISABLE; // Sync A and B disable
133     EPwm1Regs.TBCTL.bit.HSPCLKDIV   = TB_DIV1;        // Clock ratio to PWM_CLOCK_FREQ
134     EPwm1Regs.TBCTL.bit.CLKDIV      = TB_DIV1;
135
136     // Setup shadowing
137     EPwm1Regs.CMPCTL.bit.SHDWAMODE  = CC_SHADOW;
138     EPwm1Regs.CMPCTL.bit.SHDWBMODE  = CC_SHADOW;
139     EPwm1Regs.CMPCTL.bit.LOADAMODE   = CC_CTR_ZERO;   // Load on Zero
140     EPwm1Regs.CMPCTL.bit.LOADBMODE   = CC_CTR_ZERO;
141
142     //To Use Space Vector Modulation we need to swap AQ_Set and AQ_Clear
143     EPwm1Regs.AQCTLA.bit.CAU        = AQ_CLEAR;      // Set actions for EPWM1A
144     EPwm1Regs.AQCTLA.bit.CAD        = AQ_SET;
145     EPwm1Regs.AQCTLB.bit.CBU        = AQ_CLEAR;      // Set actions for EPWM1B
146     EPwm1Regs.AQCTLB.bit.CBD        = AQ_SET;
147
148     EPwm1Regs.DBCTL.bit.OUT_MODE     = DB_DISABLE;    //Disable Dead-band module
149
150     // Trip-Zone
151     //EPwm1Regs.TZSEL.bit.OSHT1      = TZ_ENABLE;     // enables TZ1 as a one-shot event source for ePWM1
152     EPwm1Regs.TZCTL.bit.TZA         = TZ_FORCE_LO;   // Force EPWMxA to a low state
153     EPwm1Regs.TZCTL.bit.TZB         = TZ_FORCE_LO;   // Force EPWMxB to a low state
154     EPwm1Regs.TZFRC.bit.OST         = 0x1;           // Software Force Trip Zone
155 }

```

Figure 59 : Init des EPWM's

Le code va ensuite suivre les étapes vues dans les Figure 57 et Figure 58. Mais la partie importante est fonction process\_control\_fast qui gère toute la partie de régulation du système, à chaque interrupt.

Conversion des mesures dans le système dq :

```

283     //Transform the 3 grid currents into dq values
284     control.dq.transformation_Grid_To_dq_Currents.in_r = measure.i1;
285     control.dq.transformation_Grid_To_dq_Currents.in_s = measure.i2;
286     control.dq.transformation_Grid_To_dq_Currents.in_t = measure.i3;
287     control.dq.transformation_Grid_To_dq_Currents.in_cos = pll.cosinus;
288     control.dq.transformation_Grid_To_dq_Currents.in_sin = pll.sinus;
289     //Calculate dq values
290     calc_rst_to_dq_transf(&control.dq.transformation_Grid_To_dq_Currents);
291     //Save the values of Id and Iq of the grid
292     control.dq.Id_grid = control.dq.transformation_Grid_To_dq_Currents.out_d;
293     control.dq.Iq_grid = control.dq.transformation_Grid_To_dq_Currents.out_q;

```

Figure 60 : Conversion des mesures dans le système dq

Il y a également une conversion pour la tension de cette même manière.

Avec les valeurs converties le système est régulé. Suivi par la normalisation des signaux et ensuite la préparation (Chapitre 6) pour envoyer les signaux sur les modulateurs.

```

404         //Set the values into the modulator. Either with TEST_PWM or with REG_ON.
405         //Phase 1
406         epwm.modulator_1.in_modulA = control.dq.output_gen1A_positive_only;
407         epwm.modulator_1.in_modulB = control.dq.output_gen1B_positive_only;
408         //Phase 2
409         epwm.modulator_2.in_modulA = control.dq.output_gen2A_positive_only;
410         epwm.modulator_2.in_modulB = control.dq.output_gen2B_positive_only;
411
412         //Phase 3
413         epwm.modulator_3.in_modulA = control.dq.output_gen3A_positive_only;
414         epwm.modulator_3.in_modulB = control.dq.output_gen3B_positive_only;

```

Figure 61 : Enregistre les valeurs dans les modulateurs

Les valeurs enregistrées sur in\_modulA et in\_modulB vont être normalisé et comparé avec la 'carrier frequency' dans les modulateurs. (Chapitre 6)

```

868 void process_modulator_AB(struct modulator_t *const modulator)
869 {
870     Uint16 tempA;
871     Uint16 tempB;
872
873     // 1 PU Limitation on input
874     LIMIT(modulator->in_modulA,0.0,1.0);
875     LIMIT(modulator->in_modulB,0.0,1.0);
876
877     // float -> pwm adaptation
878     tempA = (Uint16) (modulator->in_modulA*PWR_IN_CNT_PRD);
879     tempB = (Uint16) (modulator->in_modulB*PWR_IN_CNT_PRD);
880     // Pulse Limiter min value
881     if (tempA <= modulator->pwm_min) tempA = 0;
882     if (tempB <= modulator->pwm_min) tempB = 0;
883
884     // Pulse Limiter max value
885     if (tempA >= modulator->pwm_max) tempA = (PWR_IN_CNT_PRD);
886     if (tempB >= modulator->pwm_max) tempB = (PWR_IN_CNT_PRD);
887
888     // loading in compare register
889
890     modulator->reg_handle->CMPA.bit.CMPA = tempA;
891     modulator->reg_handle->CMPB.bit.CMPB = tempB;
892
893 }

```

Figure 62 : Fonction du modulateur

Mesures ADC pour les tensions, courants et température.

```

408     measure.u_bus_sup = ((float) *adc_val.C2 - ADC_HALF_SCALE) * ADC_TO_UBUS;
409     measure.u_bus_inf = ((float) *adc_val.C3 - ADC_HALF_SCALE) * ADC_TO_UBUS;
410     measure.u_bus = measure.u_bus_sup + measure.u_bus_inf;
411
412     measure.i1 = ((float) *adc_val.A0 - ADC_HALF_SCALE) * ADC_TO_ILEG;
413     measure.i2 = ((float) *adc_val.A1 - ADC_HALF_SCALE) * ADC_TO_ILEG;
414     measure.i3 = ((float) *adc_val.B0 - ADC_HALF_SCALE) * ADC_TO_ILEG;
415
416     measure.u1 = ((float) *adc_val.A2 - ADC_HALF_SCALE) * ADC_TO_LEG_TO_MID;
417     measure.u2 = ((float) *adc_val.A3 - ADC_HALF_SCALE) * ADC_TO_LEG_TO_MID;
418     measure.u3 = ((float) *adc_val.B2 - ADC_HALF_SCALE) * ADC_TO_LEG_TO_MID;
419
420     measure.temp_power_rack = ((float) *adc_val.C4) * ADC_TO_TEMP;

```

Figure 63 : Mesure ADC pour les courants, tensions et température

## 10.2.2 Code Space Vector Modulation

Le code pour la space vector modulation se trouve en annexe (19.10).

### 10.3 Interface HTML DSP Control

L'interface permet de contrôler le système en choisissant des paramètres.

Les paramètres disponibles sont les courants  $i_d$  et  $i_q$ . Ces courants correspondent aux courants de sortie de la partie active et réactive.

Pour un courant de  $i_d = 5\text{ A}$  et  $i_q = 0\text{ A}$  l'amplitude en sortie sur les 3 phases est de  $\sqrt{i_d^2 + i_q^2} = 5\text{ A}$  avec uniquement de la puissance active. En changeant les valeurs d' $i_d$  et  $i_q$  il est possible de changer le courant délivré avec des parties de puissances actives et réactives différentes.

Mais cette page ne permet pas d'uniquement paramétrer les courants  $i_d$  et  $i_q$  mais également de voir l'état du système et de le piloter. Il est même possible d'observer les valeurs des variables du code.

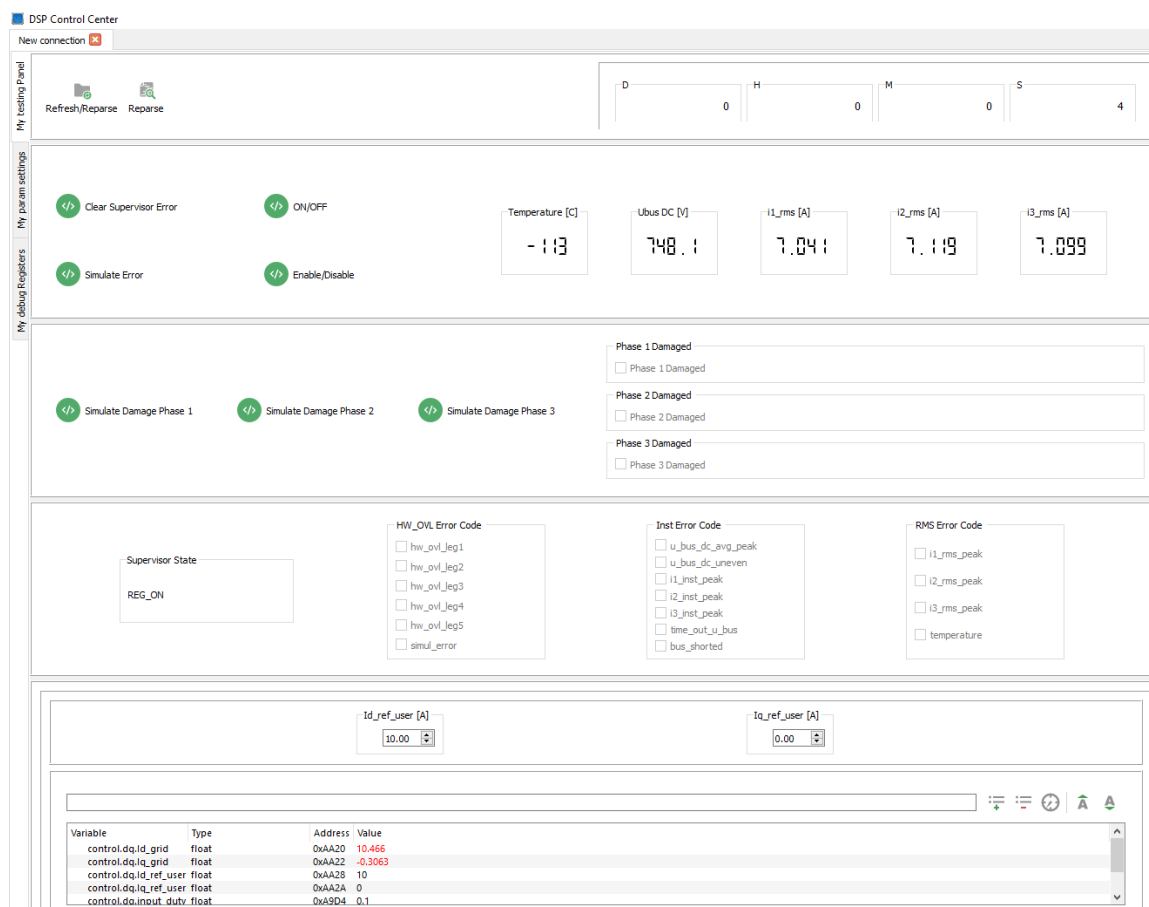


Figure 64 : Interface HTML DSP Controller

### 10.4 Remarque CPLD Poetic

La CPLD Poetic est la même que pour les autres projets POETIC. Elle s'occupe juste d'envoyer les signaux du CPU vers les bons modules d'entrée ou sorties. Cependant il y a une modification à faire sur les sorties PWM. La programmation précédente bloque les sorties PWM, si la PWM\_A est activé cela bloque l'activation de la PWM\_B. Il est nécessaire de désactiver cette sécurité pour pouvoir faire la transmission des signaux selon le chapitre 6. Car PWM\_A et PWM\_B doivent pouvoir être activé en même temps pour le projet ANPC.

## 11 Simulation Typhoon HIL (Source DC + ANPC + Injection Réseau)

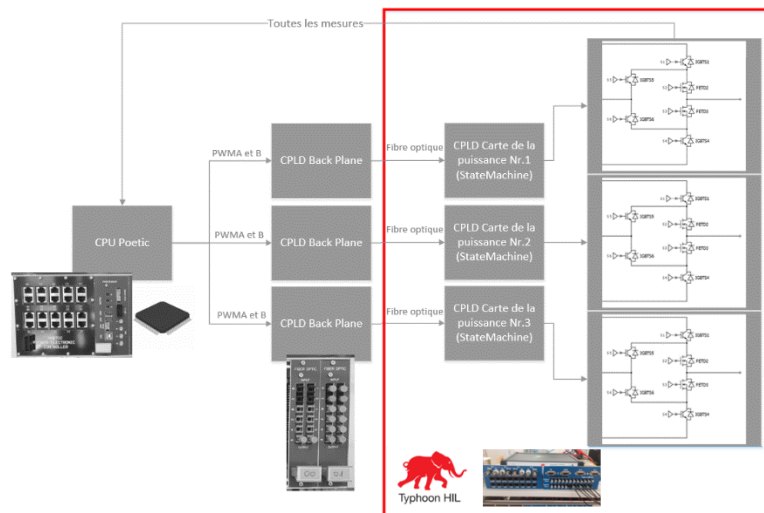


Figure 65 : Dans ce chapitre les CPLD des cartes de puissance et les semi-conducteurs sont remplacé par une simulation temps réel avec Typhoon HIL. La seule partie traitée dans ce chapitre est la simulation Typhoon ou la partie en couleur de l'image

### 11.1 Schéma

La simulation Typhoon HIL permet de tester le système sans utiliser la carte de puissance réel. Ceci permet de faire des tests du système sans danger et sans détruire les semi-conducteurs en allant aux limites du fonctionnement.

La simulation contient les alimentations DC nécessaires, un réseau 50Hz où l'installation vient se connecter, les semi-conducteurs en topologie ANPC (Mosfet et ou IGBT) ainsi que la state machine de commande qui est sur cette carte réelle. La state machine est la PWM4 disponible dans le chapitre 5.1 sans les états de protection.

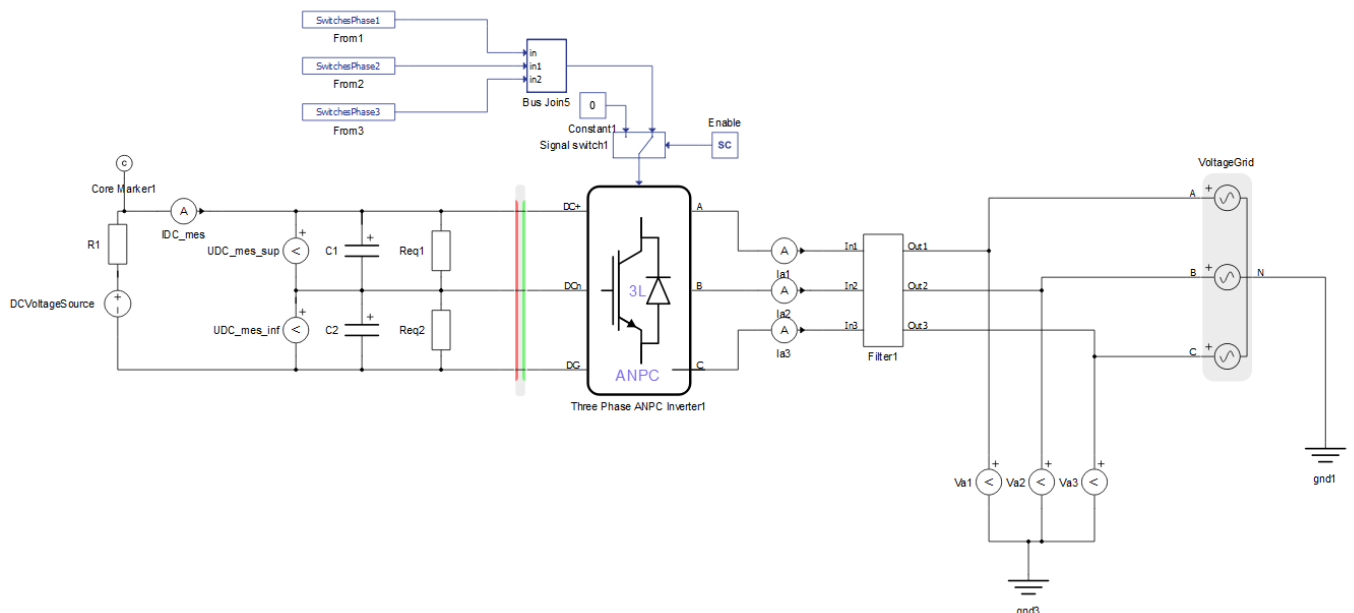


Figure 66 : Schéma de simulation Typhoon HIL



## 11.2 Contrôle de l'installation

Commande du système : Le système Poetic pilote le système comme pour le fonctionnement réel. Poetic ne peut pas différencier entre une simulation et le système réel. Les signaux sont envoyés par fibre optique sur des entrées de Typhoon. Pour commander le système il y a 4 états de 0 à 3 codés sur 2 fibres optiques. Le signal doit être décodé sur la simulation Typhoon HIL. Sur la Figure 67 pour la première phase il y a un Digital Input 1 (PWM1\_A) et Digital Input 2 (PWM1\_B). Input 1 à un gain de 2 et Input 2 à un gain de 1 et ensuite les deux valeurs sont additionnées. Pour les trois phases il faut lire les inputs et y appliquer un gain de 1 ou de 2.

La somme de ce signal est une entrée de la machine d'état (Machine d'état de la Figure 13). La machine d'état utilise cette entrée et va fournir 7 sorties. 6 sorties indiquent comment ouvrir ou fermer les semi-conducteurs dans l'onduleur ANPC et une dernière sortie pour observer dans quel état la state machine se trouve (dépannage).

Une fois que les 18 signaux sont disponibles pour contrôler le système, ils sont branchés sur l'onduleur, visible à la Figure 66.

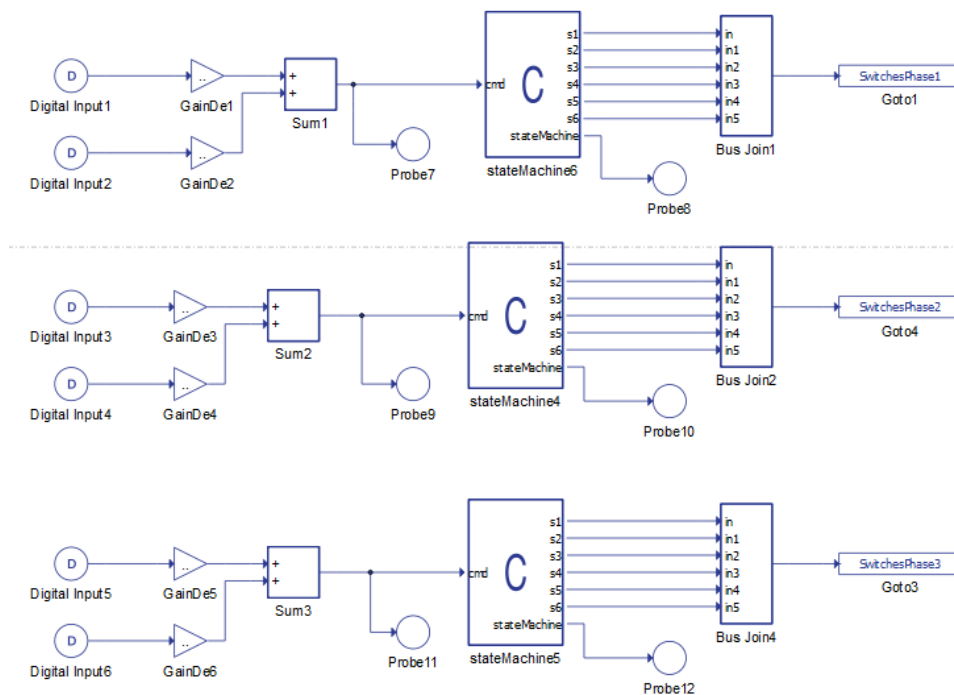


Figure 67 : Lecture des entrées fibres optiques et state machine

## 11.3 Communication avec Poetic

En remplaçant une partie du système électronique par une simulation Typhoon HIL, il ne sera pas possible d'envoyer des signaux depuis la carte physique. Mais sur Typhoon HIL il y a des sorties analogiques qui permettent de remplacer ces mesures manquantes.

Deux valeurs de tension DC, qui correspondent à la tension sur le condensateur haut et à la tension sur le condensateur bas, sont nécessaires. Ces deux tensions ensemble permettent d'avoir la tension totale sur le bus DC. Dans certains cas de fonctionnement les tensions DC supérieur et inférieur peuvent diverger en gardant une tension totale normale. Ceci permet d'avoir un contrôle plus précis sur le fonctionnement. Chapitre 12.6 sur l'équilibrage des tensions DC.

Il y a également les 3 tensions de réseaux, qui permettent de mesurer l'amplitude de la tension de réseau mais surtout pour introduire une PLL. Une PLL sert à connaître la fréquence du réseau et ainsi se mettre en phase avec l'onduleur. Les trois dernières mesures sont les 3 valeurs de courants qui sortent de l'onduleur. Elles sont nécessaires pour réguler le système à un courant de référence, mais aussi de faire un déphasage lorsque la consigne impose des puissances actives et réactives spécifiques.

Pour que Typhoon fournissent ces valeurs sur des sorties analogiques et les récupérer sur Poetic il faut paramétrer les sorties analogiques Typhoon. (Figure 68)

Output	Signal	Scaling	Offset (Vdac)	Lower/Upper Limit (Vdac)	Lock
AO1	UDC_mes_sup	95.703 V per 1 Vdac	0.0	-10.0 10.0	
AO2	Ia1	13.333 A per 1 Vdac	0.0	-10.0 10.0	
AO3	Ia2	13.333 A per 1 Vdac	0.0	-10.0 10.0	
AO4	Ia3	13.333 A per 1 Vdac	0.0	-10.0 10.0	
AO5	Va1	46.231 V per 1 Vdac	0.0	-10.0 10.0	
AO6	Va2	46.231 V per 1 Vdac	0.0	-10.0 10.0	
AO7	Va3	46.231 V per 1 Vdac	0.0	-10.0 10.0	
AO8	UDC_mes_inf	95.703 V per 1 Vdac	0.0	-10.0 10.0	

Figure 68 : Sorties Typhoon HIL de tension et courant. 3x Courants de réseau, 3x Tensions de réseau et 2 x Tensions BUS DC qui correspondent aux valeurs DC Condensateur supérieur et inférieur

La valeur de 'Scaling' des sorties analogiques permet d'imposer un gain sur la sortie. Le gain dépend de la valeur max de la mesure.

Pour les courants le gain est de 13.333.

Pour les tensions < 450 V le gain est de 46.231.

Pour les tensions < 900V le gain est de 95.703. La tension sur les condensateurs n'est pas plus grande que 450V mais un déséquilibre dans le système entre les deux tensions du BUS DC peut faire augmenter/baisser les tensions. Le gain choisi est celui de < 900 V.

Ces gains correspondent aux valeurs calculées par Monsieur Gabriele Pescatore pour le projet Moteur U/f. [3]

Configuration	HIL Scale			
	RJ45 FS	GainBuffer dac out	Scale V/V	Offset
120V standard	1.667	0.1667	12.346	0
	0.833	0.1667	12.346	0
450V standard	1.667	0.1667	46.231	0
	0.833	0.1667	46.231	0
900V standard	1.667	0.1667	95.703	0
	0.833	0.1667	95.703	0

Configuration	HIL Scale			
	RJ45 FS	GainBuffer dac out	Scale A/V	Offset
Série 25 standard	1.666667	0.1667	6.667	0
	0.833333	0.1667	6.667	0
Série 25 standard	1.666667	0.1667	2.222	0
	0.833333	0.1667	2.222	0
Série50 standard	1.666667	0.1667	13.333	0
	0.833333	0.1667	13.333	0

Figure 69 : Valeur des gains pour les sorties Typhoons HIL, Travail de Bachelor Laboratoire U/f [3]

### 11.4 Panel de commande Typhoon HIL

Le panel de commande Typhoon permet d'observer des signaux, mais aussi de piloter l'installation dans des états spéciaux. Les signaux de commande du système arrivent depuis Poetic vers Typhoon lorsque l'installation est activée, mais il est possible de couper la régulation dans ce panel. On peut activer la tension de réseau ou même y faire varier l'amplitude et la fréquence pour faire des tests de la PLL.

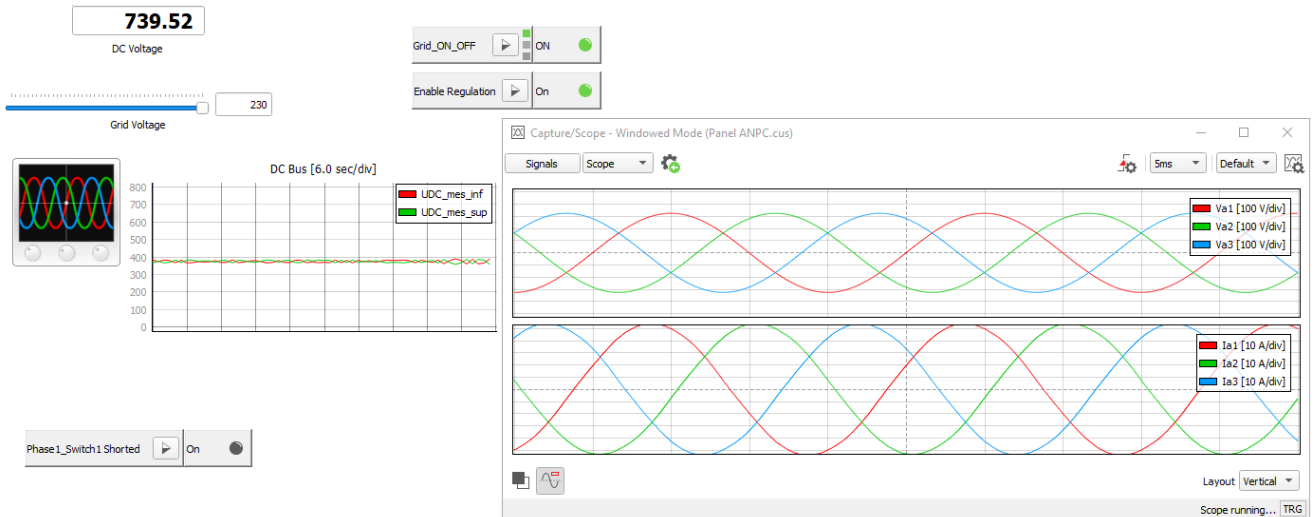


Figure 70 : Panel ANPC de visualisation et de commande de l'installation

## 12 POETIC + Simulation Typhoon

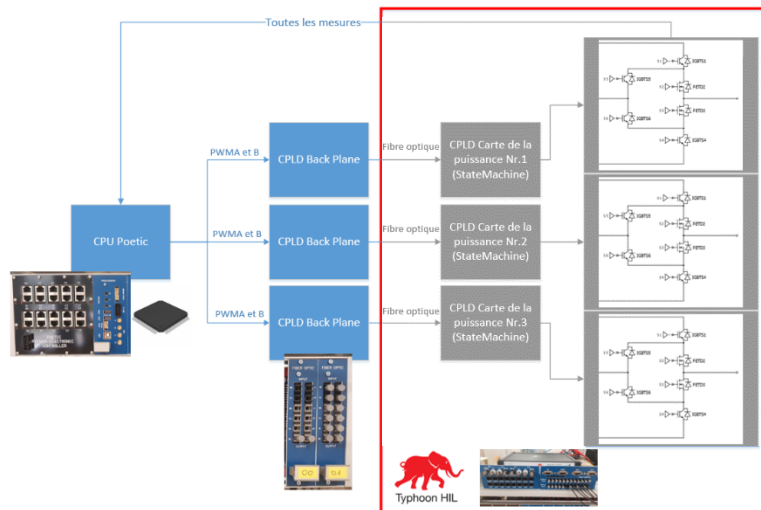


Figure 71 : Ce chapitre traite des simulations du système avec Typhoon HIL sans les cartes de puissance réel

Une fois que les différentes parties fonctionnent individuellement il sera possible de tout lier. Les parties importantes de ces tests sont les moyens de communication entre les différents systèmes. Pour envoyer les informations de Typhoon vers Poetic avec les câbles Ethernet il faut suivre la configuration avec le chapitre 11.3 et les branchements sont disponibles en annexe 19.7

## 12.1 Contrôle du temps d'échantillonnage Typhoon

Pour commencer les tests il faut passer étape par étape et corriger les problèmes les un après l'autre. Typhoon ne pouvait pas lire tous les signaux fibres optiques et en augmentant la vitesse d'échantillonnage une fenêtre d'erreur est apparue. Le Typhoon HIL choisi n'était pas assez performant il n'arrivait pas à suivre (trop de calculs). Il a fallu choisir un HIL plus performant (HIL404, Noir).

## 12.2 Contrôle de la conversion des signaux fibres optiques

Avec le bon HIL choisi les premiers tests peuvent commencer avec les mesures sur les fibres optiques. Suivi par les signaux fibres optiques et la conversion en signaux pour la state machine. Typhoon possède un oscilloscope et les valeurs peuvent être observé sur le programme directement.

Signaux sur la fibre optique avec la modulation :

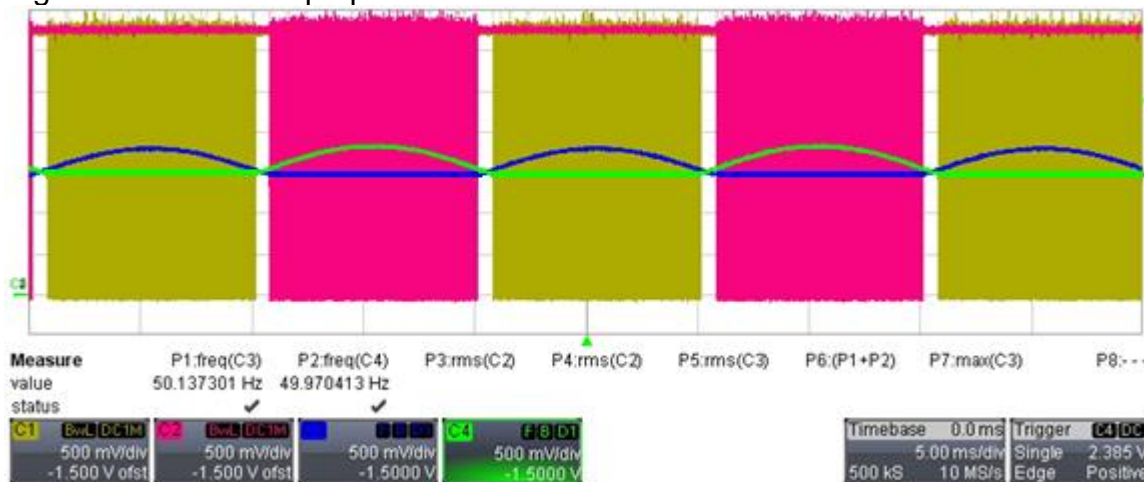


Figure 72 : Valeur de modulation pour la phase 1 sur A et B avec les signaux C3 et C4. Sortie du signal fibre optique qui sera envoyé vers Typhoon Hil avec les signaux C1 et C2.

Les signaux fibres optiques que Typhoon arrive à lire.

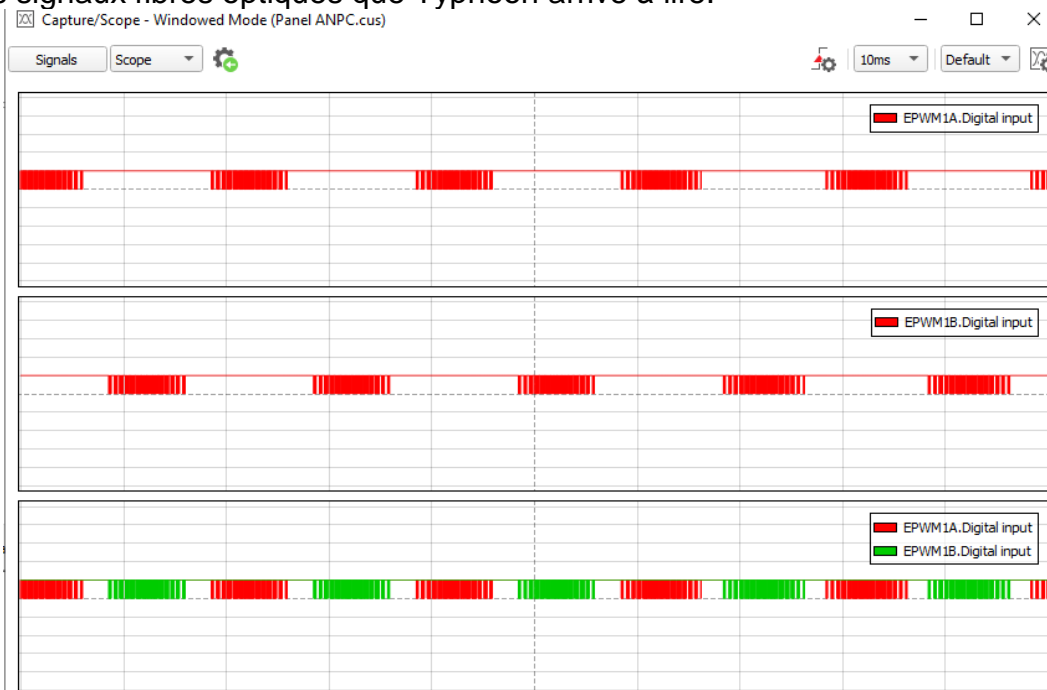


Figure 73 : Signaux que reçoit Typhoon Hil avec les fibres optiques. Les 2 premiers graphiques les signaux A et B individuel et le 3ème les deux signaux ensemble.

## Décodage des signaux reçu par Typhoon comme vu au chapitre 6.4



Figure 74 : Décodage des signaux reçu. Le signal A à un gain de 2 et le signal B à un gain de 1. La somme des deux donne une valeur entre 0 et 3 sur le deuxième graphique.

Le signal codé va piloter la machine d'état. Le premier graphique correspond au signal codé. Sur le deuxième graphique il y a l'état où la state machine se trouve. (State machine PWM4 avec des états entre 1 et 5).



Figure 75 : Le signal décodé vient sur la state machine et en sortant de cette state machine nous allons avoir des sorties activées ou non mais également un numéro d'état entre 1 et 5. Voir chapitre 5.1

Le décodage et la state machine fonctionnent. Il est maintenant possible d'envoyer ces signaux sur l'onduleur ANPC de Typhoon pour voir son fonctionnement.

### 12.3 Test installation Typhoon

#### 12.3.1 Open-Loop

Juste en envoyant des signaux sur l'onduleur sans faire de régulation il est possible de faire des tests.

Signaux qui sont envoyés :

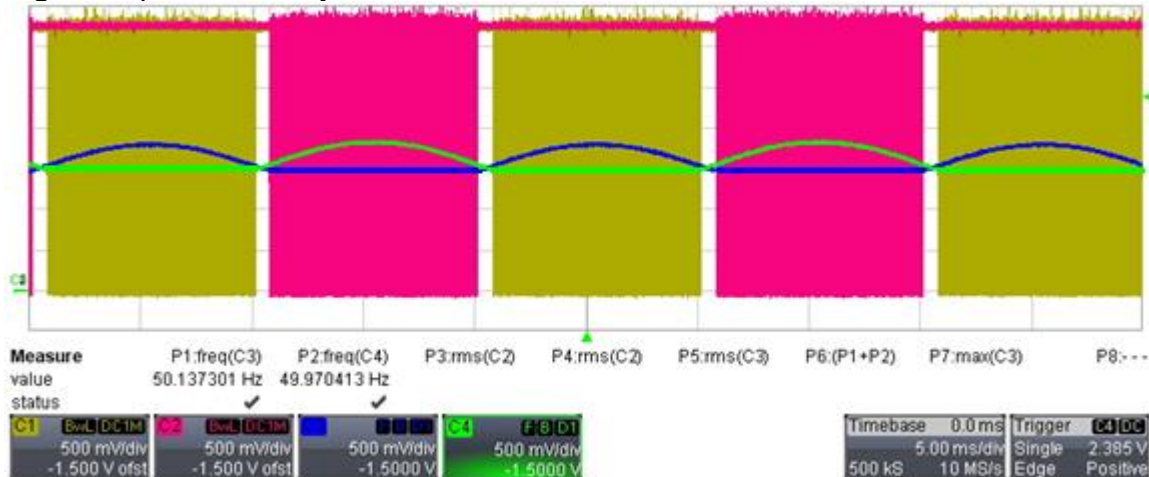


Figure 76 : Signaux qui sont envoyés pour faire fonctionner l'onduleur en open-loop. Même image que la Figure 72.

Lorsque la state machine en rouge est dans la partie supérieur il faut avoir une tension positive sur le signal vert et inversement.

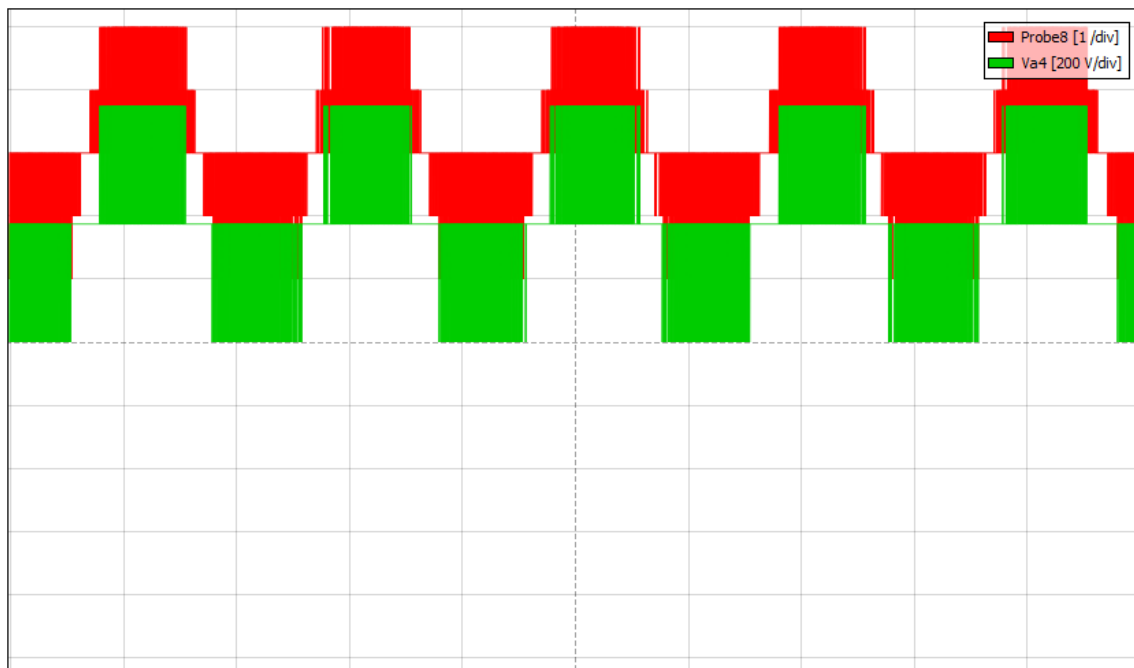


Figure 77 : Tension en sortie de l'onduleur en vert avec l'état de la state machine en rouge.

Test de la tension sur les trois phases :



Figure 78 : Tension en sortie de l'onduleur pour les trois phases. ( $120^\circ$  entre les phases)

Test de la tension de réseau en lien avec la tension de l'onduleur.

Les deux ne sont pas en phase, c'est normal car le programme n'est pas synchronisé avec la PLL en open-loop.

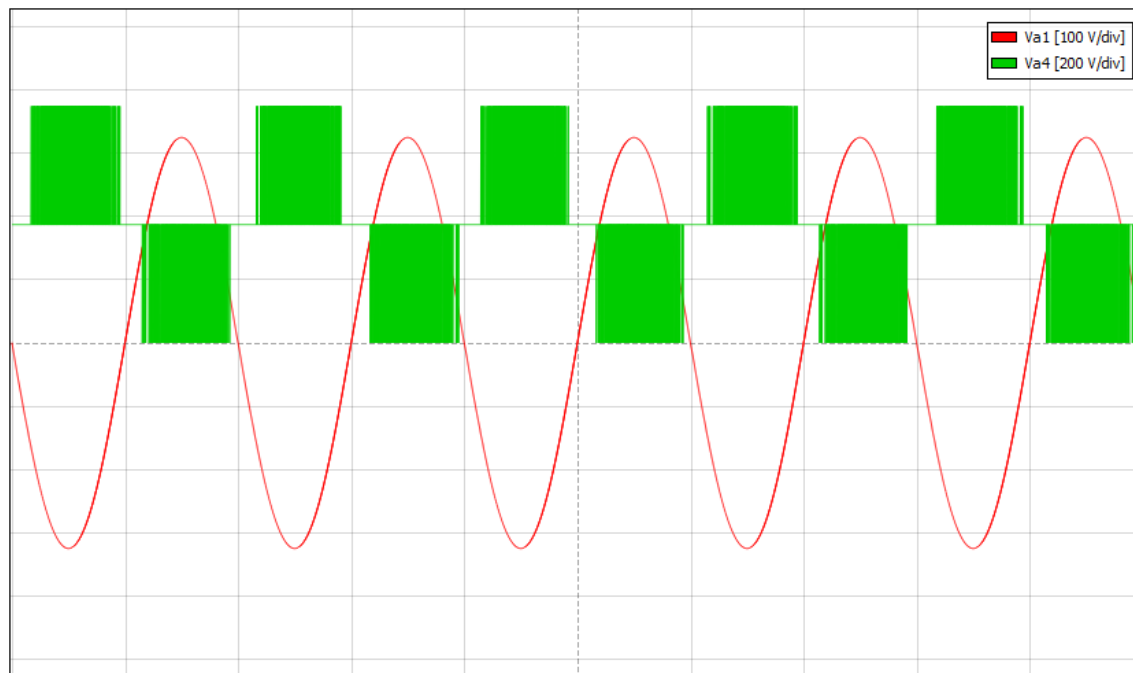


Figure 79 : Lors des tests PWM la sortie de l'onduleur n'est pas encore synchronisé sur le réseau. Les sorties ne sont pas en phase.

Une fois le résultat voulu obtenu il est possible de connecter le filtre et/ou le réseau pour y faire passer un courant dans la simulation Typhoon.



### 12.3.2 Closed Loop (Grid off)

En sortie du système il y a uniquement une inductance de 1mH, pas de résistance. Pour avoir un meilleur sinus il faut augmenter la valeur de l'inductance ou aussi ajouter un filtre.

En closed-Loop il est possible de choisir le courant que l'utilisateur veut en sortie. Sur l'interface DSP Controller (chapitre 10.3) l'utilisateur choisi un courant  $I_d$  et  $I_q$  et l'installation va se réguler pour obtenir ces valeurs.

En système closed-Loop sans le réseau les paramètres  $I_d$  et  $I_q$  n'ont pas d'importance car il n'y a pas de tension à comparer pour y crée le déphasage. Mais pour l'amplitude du courant complet la formule reste  $\sqrt{i_d^2 + i_q^2}$ .

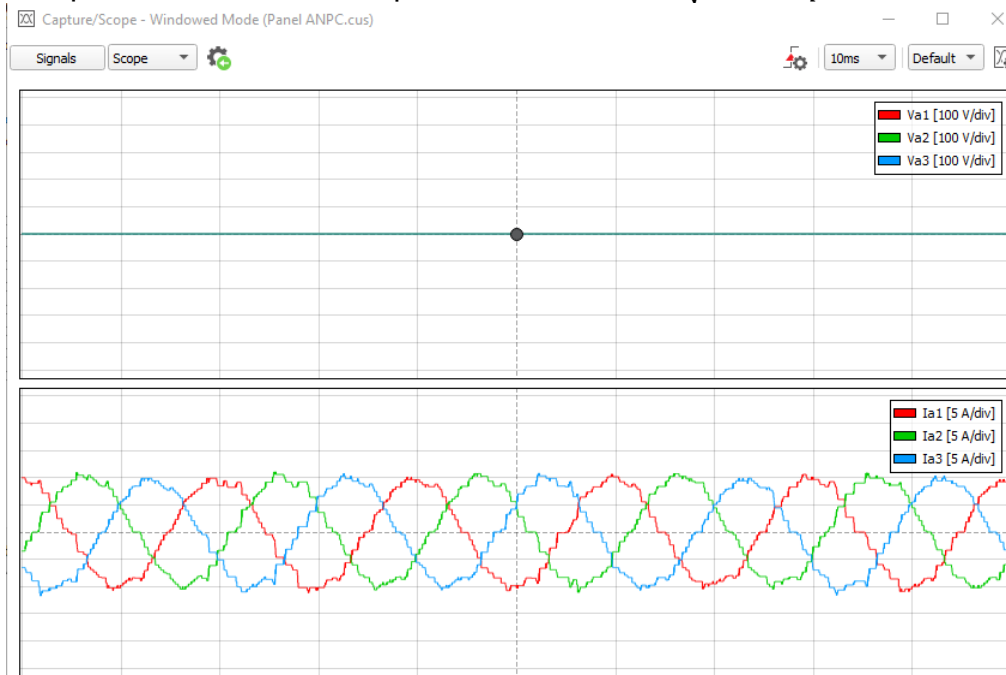


Figure 80 : Système en closed-Loop sans tension de réseau. Régulation sur 10 A

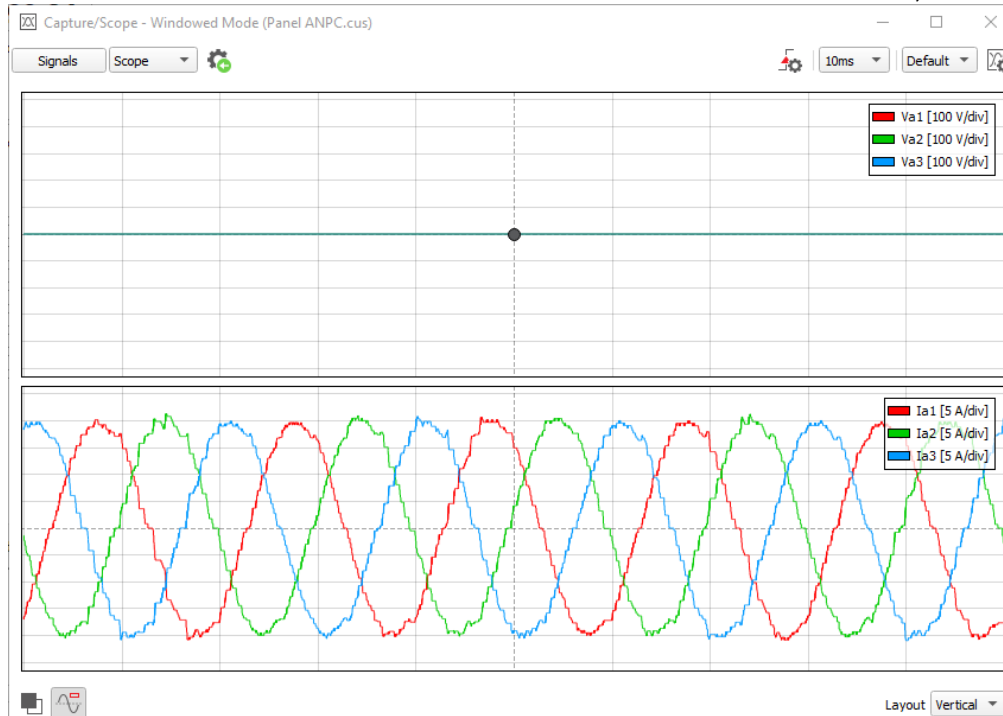


Figure 81 : Système en closed-Loop sans tension de réseau. Régulation sur 20 A

### 12.3.3 Closed-Loop (Grid On)

En closed-Loop il est possible de choisir le courant que l'utilisateur veut en sortie. Sur l'interface DSP Controller (chapitre 10.3) l'utilisateur choisi un courant  $I_d$  et  $I_q$  et l'installation va se réguler pour obtenir ces valeurs.

Si le réseau est activé les paramètres  $I_d$  et  $I_q$  sont importants car il est possible de ne transmettre uniquement de la puissance active en gardant  $I_q$  à 0. Mais si nécessaire on peut choisir de transmettre de la puissance active et ou réactive.

Quelques exemples :

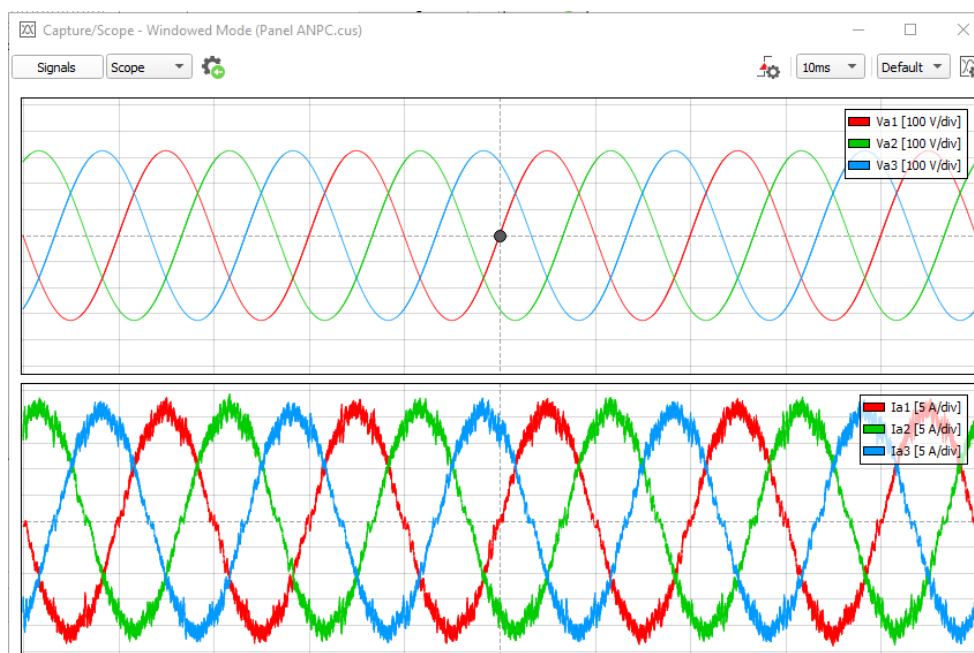


Figure 82 : Courant voulu :  $I_d = 20$  A et  $I_q = 0$  A. L'utilisateur ne veut pas de puissance réactive. Le courant et la tension sont en phase.

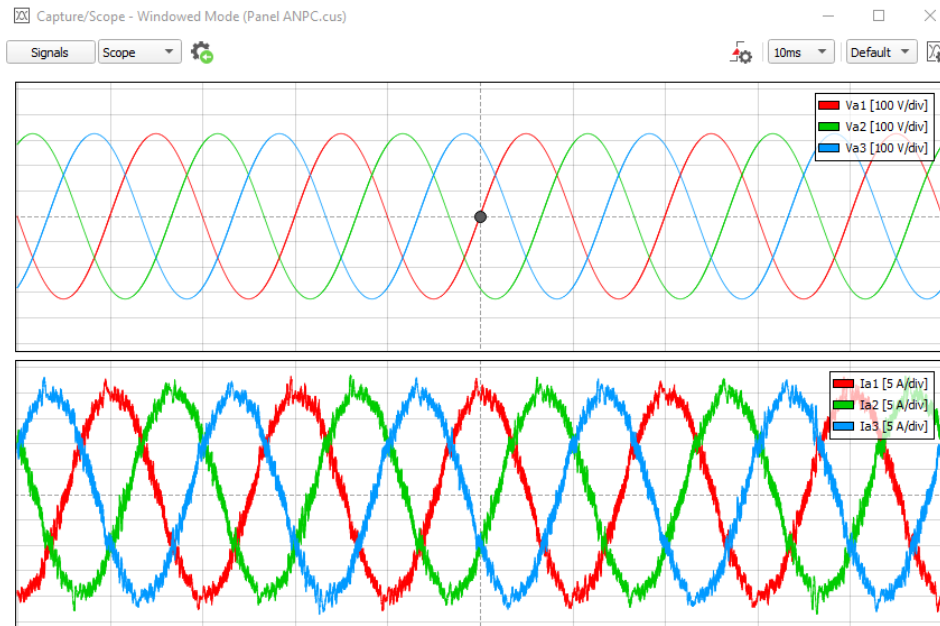


Figure 83 : Courant voulu :  $I_d = 0$  A et  $I_q = 20$  A. L'utilisateur veut uniquement de la puissance réactive. Le courant est déphasé de  $90^\circ$

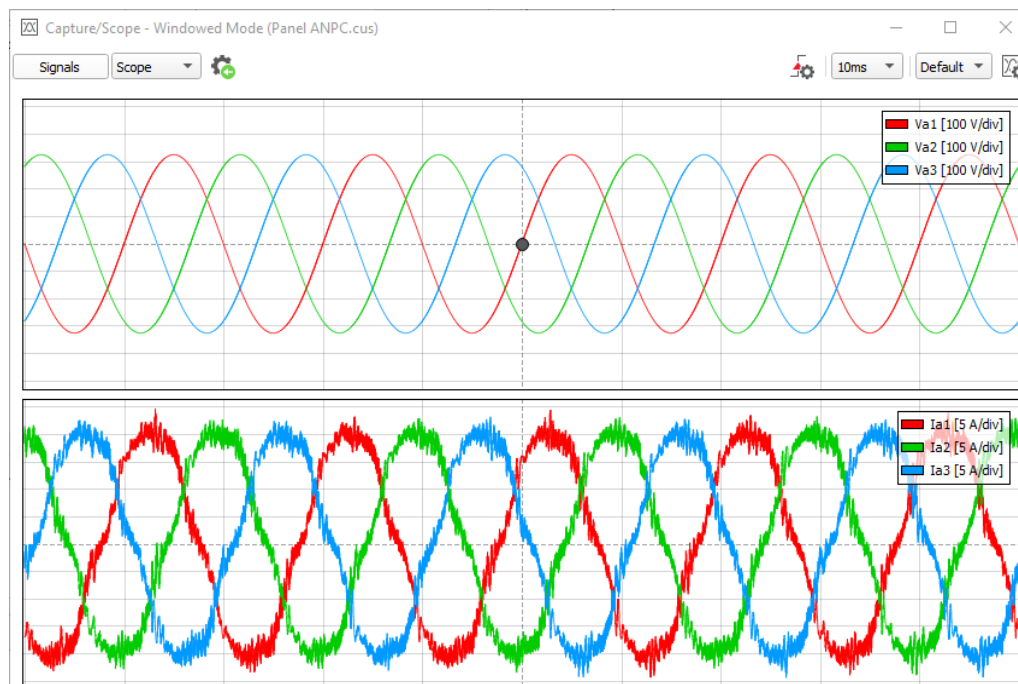


Figure 84 : Courant voulu :  $I_d = 14.14$  A et  $I_q = 14.14$  A. L'utilisateur veut autant de puissance active que réactive. Le courant est déphasé de  $45^\circ$

## 12.4 Closed-Loop sur le réseau avec une consigne à 0A

Lorsque l'onduleur est branché sur le réseau et l'utilisateur ne veut pas transmettre de puissance ou 0A en sortie il ne suffit pas de couper l'installation. La tension du réseau tourne à 50 Hz à 325V crête. Si l'onduleur reste à 0V pendant qu'il est branché il va y avoir une différence de tension de 325V sur la ligne, ce qui génère un grand courant. L'onduleur doit fournir une tension en phase avec le réseau pour ne pas générer un courant.

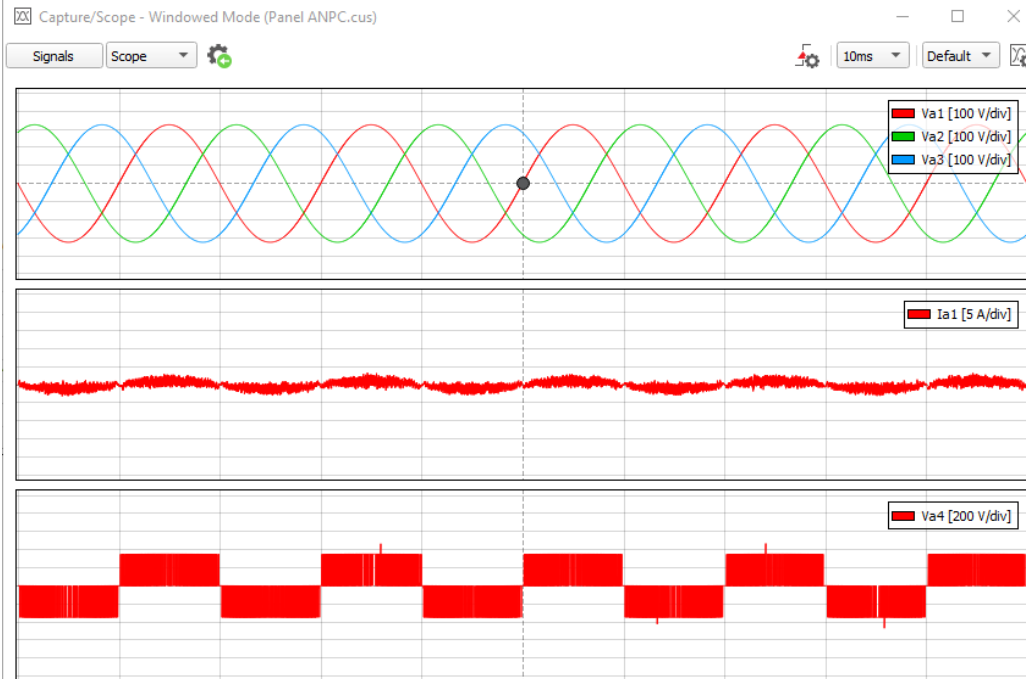


Figure 85 : Graphique 1 = Tensions de réseau, Graphique 2 = Courant transmis avec une consigne de 0A, Graphique 3 = Tension que l'onduleur doit produire pour ne pas générer de courant ou très peu. La tension n'est pas parfaitement en phase avec la bonne amplitude et nous avons un faible courant d'environ 100mA

## 12.5 Simulation Space Vector Modulation

La simulation en mode Space Vector Modulation donne des courants de sorties avec la tension de réseau.

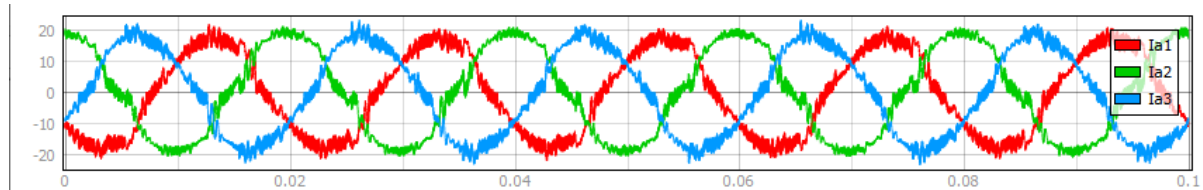


Figure 86 : Courants de sorties en mode SVPWM\_1

Il est même possible d'activer le système avec uniquement des tension positives ou négatives en sorties avec les bonnes consignes.

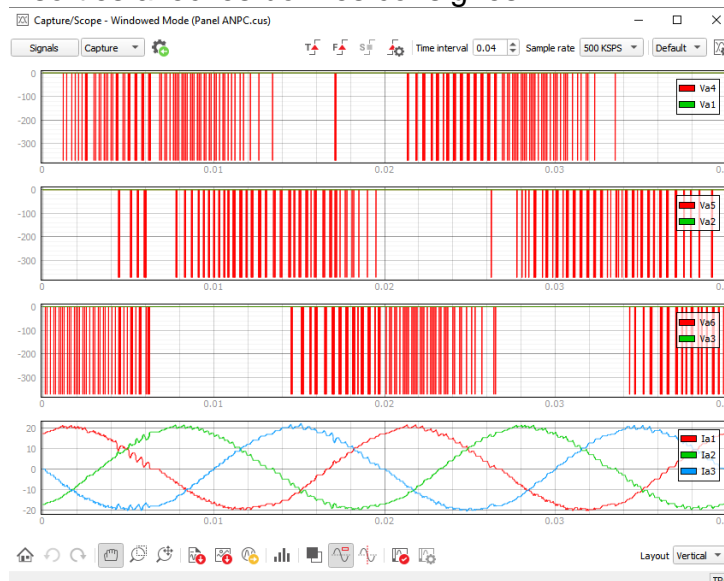


Figure 87 : Fonctionnement avec uniquement des vecteurs qui crée des tensions négatives en sorties, Graphique 1-3 = Tensions en sorties de l'onduleur, Graphique 4 = Les trois courants en sorties du système

## 12.6 Equilibrage de la tension DC

La tension totale DC est séparée en deux tension sur des condensateurs avec une tension supérieur et tension inférieur.

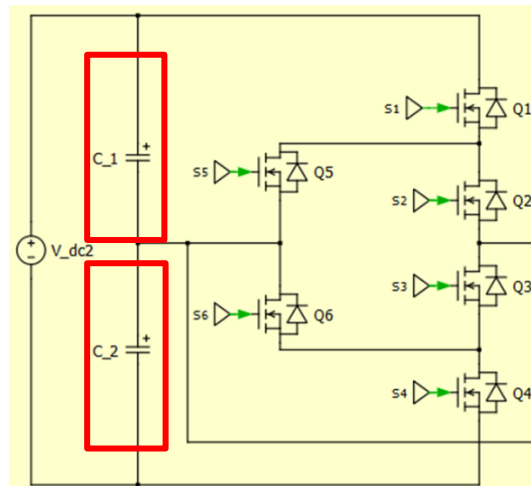


Figure 88 : Tension DC totale sur deux condensateurs

Ces deux tensions peuvent diverger tout en gardant une tension DC totale qui reste à la bonne valeur. Pour résoudre ce problème il y a deux méthodes :  
 Première méthode en SVPWM: En mode SVPWM il suffit d'utiliser les vecteurs alternatifs. Exemple : Si la tension du condensateur supérieur est plus importante il faut utiliser les vecteurs avec des 'P' pour la commande. Si c'est la tension du condensateur inférieur qui est plus importante il faut utiliser les vecteurs avec des 'N' pour la commande. Les vecteurs P consomment de la tension du condensateur supérieur et N inversement. Car il y a des vecteurs identiques comme par exemple : POO est égal à ONN, il suffit de changer le vecteur.

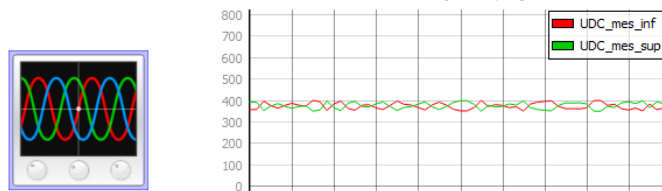


Figure 89 : Au sommet de l'image il y a la tension qui s'équilibre et sur oscilloscope il y a le courant en sortie du système

Deuxième méthode en PWM : Le sinus de modulation est entre -1 et 1. Pour la partie positive de la modulation cela consomme de la tension sur le condensateur supérieur et pour la partie négative inversement. En ajoutant un offset sur la modulation cela permet de garder la même forme pour le signal de modulation tout en contrôlant les tensions DC.

Pour contrôler l'offset une régulation est mise en place avec en entrée la différence des tensions.

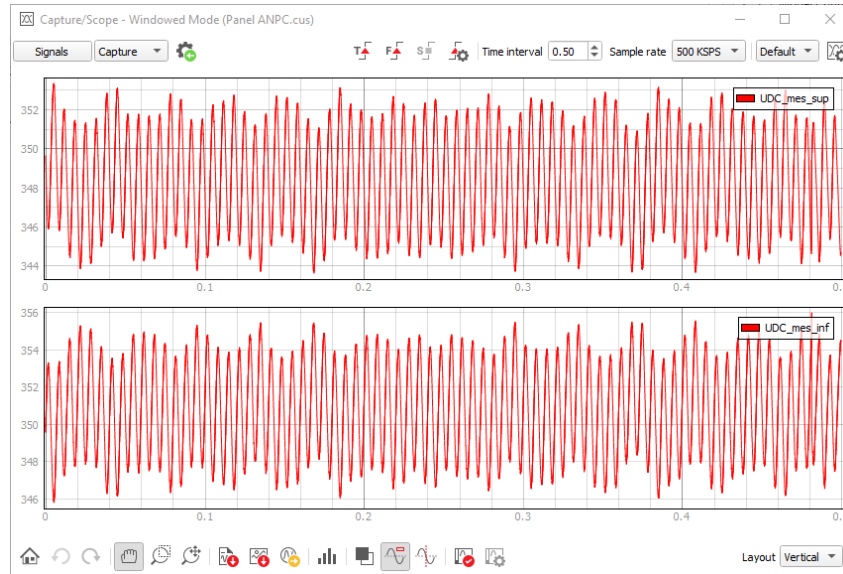


Figure 90 : Les tensions sur les deux condensateurs reste semblables avec ce contrôle (Ecart de 6V max)

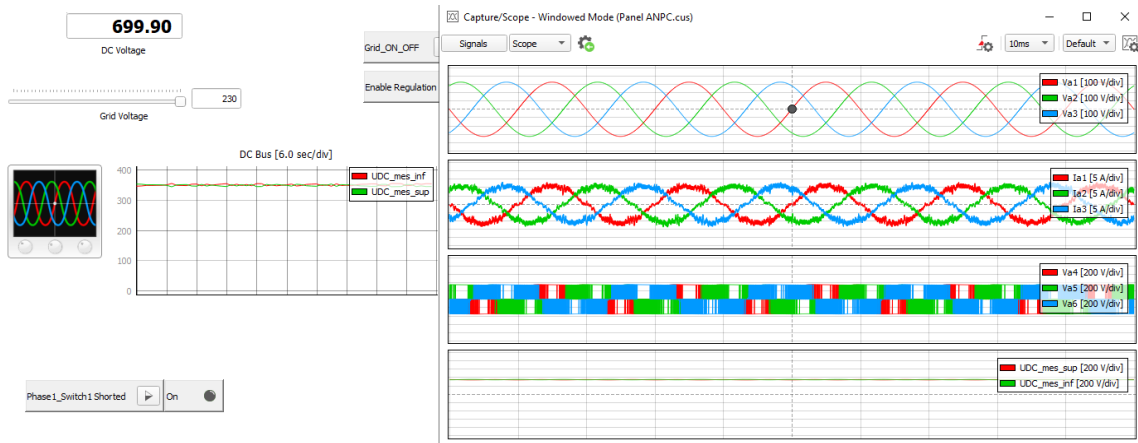


Figure 91 : A gauche de l'image les tensions sont stables avec une faible différence. Sur la droite de l'image dans le deuxième graphique il y a les courants de sortie qui sont identiques aux signaux des autres tests.

Les trois références de tensions normalisées pour les modulateurs changent beaucoup pour aider à l'équilibre des tensions DC.

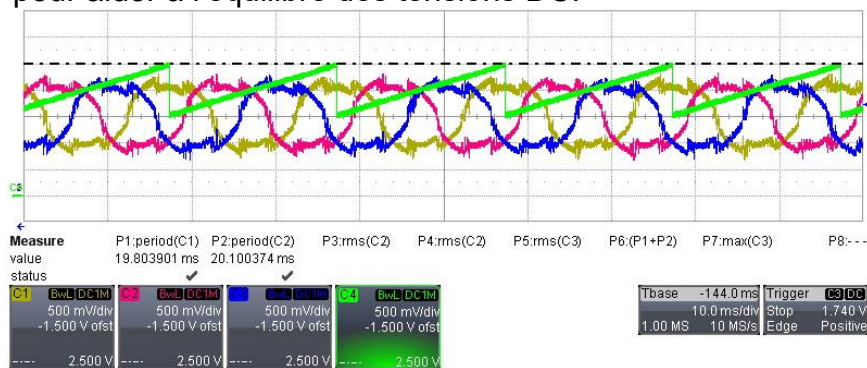


Figure 92 : C1 à C3 = Tensions de références normalisées pour les modulateurs, C4=Angle de la PLL



## 12.7 Test de performance de l'installation

### 12.7.1 Fréquence PLL

La PLL permet de calculer la fréquence du réseau, mais aussi d'avoir une référence d'angle de 0 à 360° pour savoir à n'importe quel moment où le réseau se trouve. Ceci permet de brancher le système sur le réseau.

Dans la figure suivante le test est de changer la fréquence du réseau pour observer comment la PLL réagit et à quelle vitesse.

Changement de 50Hz à 40 Hz :

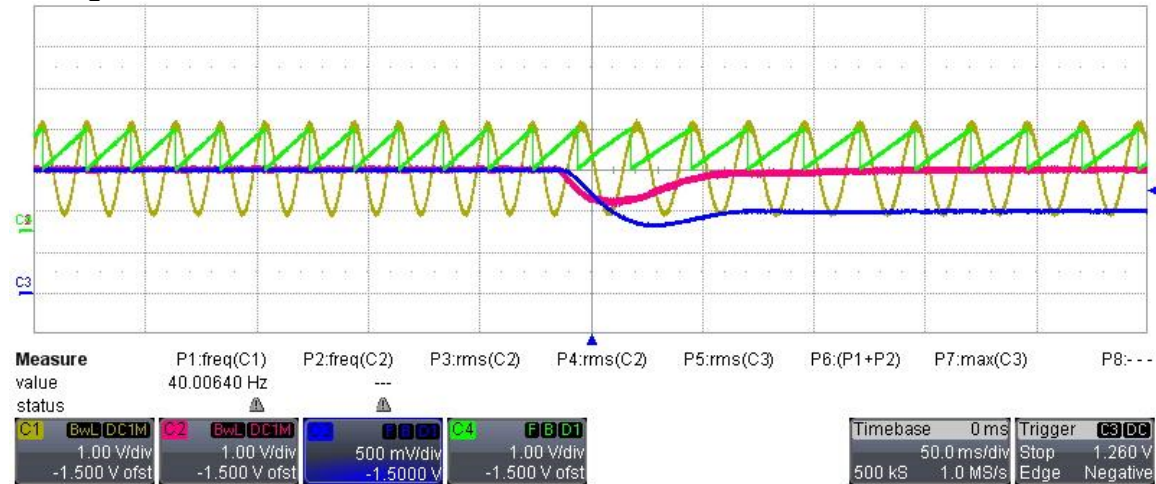


Figure 93 : Changement de fréquence de 50 à 40 Hz : C1 = Tension de réseau, C2 = Différence de fréquence sur le régulateur, C3 = fréquence du réseau (50Hz au milieu et 10Hz/dec), C4 = Angle de la PLL

Changement de 50Hz à 60Hz :

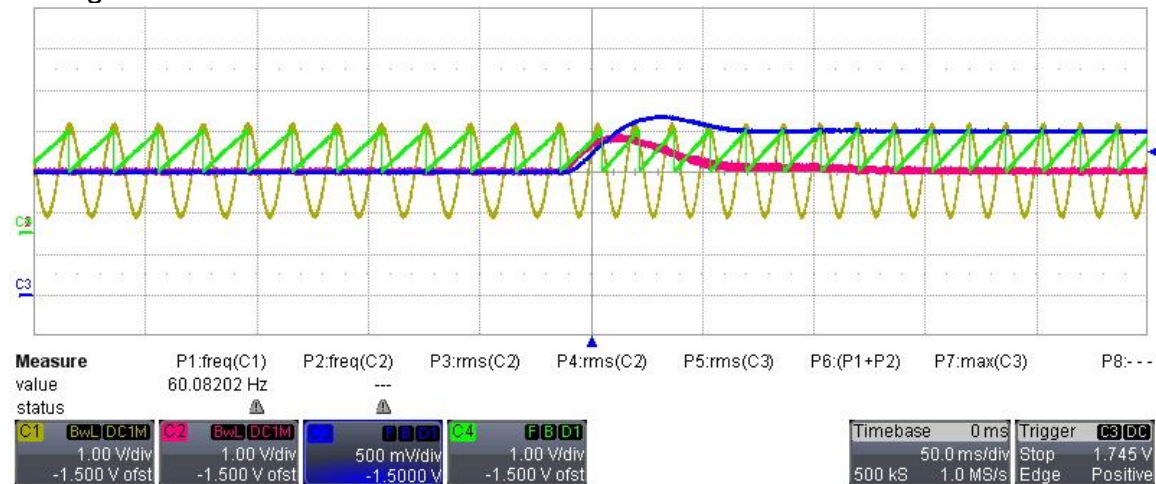


Figure 94 : Changement de fréquence de 50 à 60 Hz : C1 = Tension de réseau, C2 = Différence de fréquence sur le régulateur, C3 = fréquence du réseau (50Hz au milieu et 10Hz/dec), C4 = Angle de la PLL

Le système réagit vite, environ 50ms pour changer la valeur de la fréquence sur la PLL.



### 12.7.2 Vitesse de régulation du courant

La vitesse de régulation est importante. Dans la figure suivante il y a un saut de consigne de courant. On commence à 0A et termine à 10A.

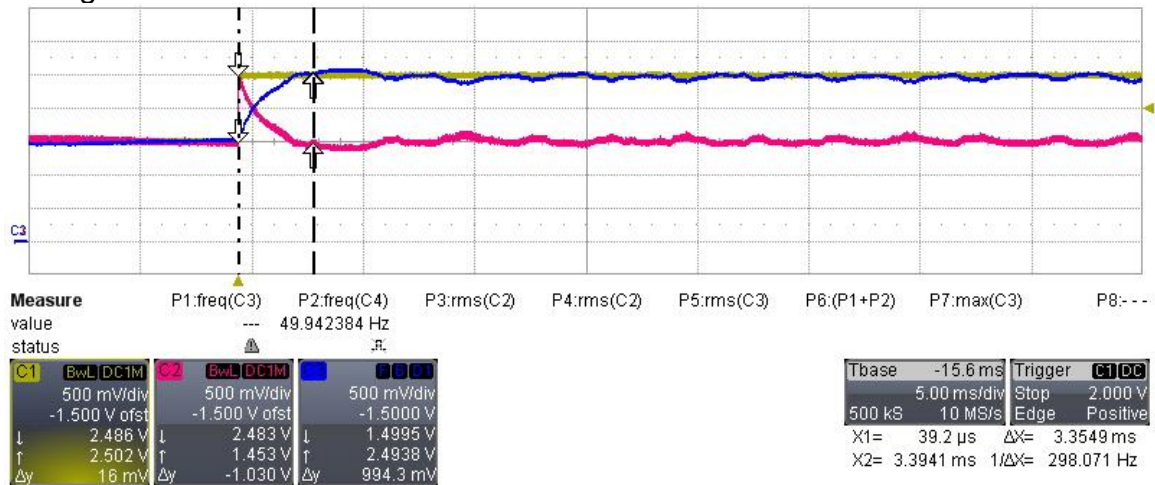


Figure 95 : Changement de consigne de courant  $I_d$ , commence à 0A et fini à 10A, C1 = Consigne de courant de l'utilisateur, C2 = Erreur de courant sur le régulateur  $I_d$ , C3 = Le courant mesuré en sortie du système. La vitesse de ce changement se fait en environ 3.4 ms.

### 12.7.3 Tension DC limite

La tension DC nécessaire pour avoir un bon fonctionnement est de 700V pour le fonctionnement PWM, car pour une tension DC plus faibles la sortie n'arrive pas à fournir une tension suffisante. (Exemple avec tension DC plus faible à la Figure 97)

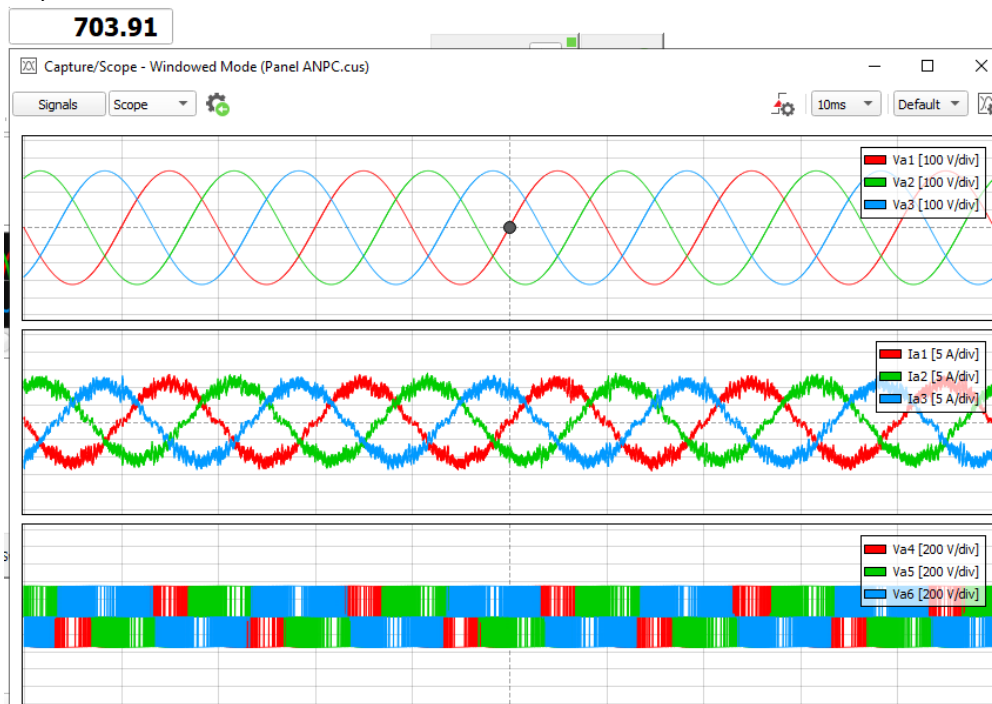


Figure 96 : Fourni 10A sur le réseau pour une Tension DC de 700V

Si la tension DC est plus faible, la sortie de l'onduleur ne peut pas fournir les 325V crête en sortie pour suivre la tension de réseau et le système est limité et fourni un courant de sortie non sinusoïdal.

652.71

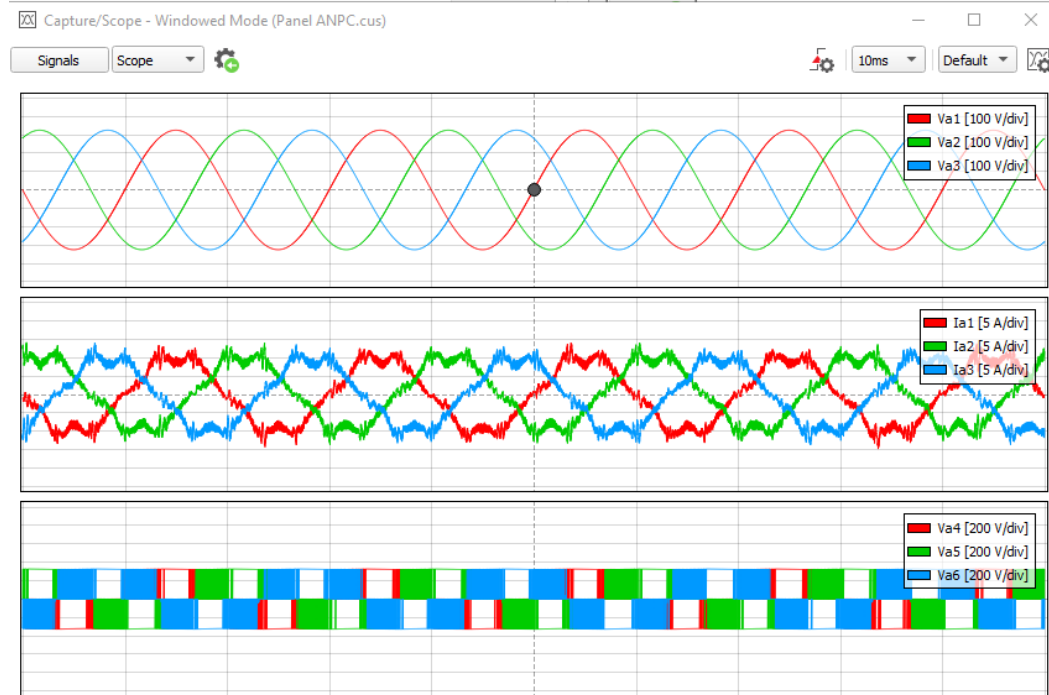


Figure 97 : Fourni 10 A sur le réseau pour une tension DC de 650V (Tension DC trop faible pour être sinusoïdal)

#### 12.7.4 Test FFT des courants pour les harmoniques

Les tests FFT permettent de contrôler si le système crée trop d'harmoniques. Mesure pour une tension de 700V et consigne de courant  $I_d$  à 10A :

La norme CEI 61000-3-2 pour des appareils de classe A :

Tableau 1 – Limites pour les appareils de classe A

Rang harmonique $n$	Courant harmonique maximal autorisé $A$
<b>Harmoniques impairs</b>	
3	2,30
5	1,14
7	0,77
9	0,40
11	0,33
13	0,21
$15 \leq n \leq 39$	$0,15 \frac{15}{n}$
<b>Harmoniques pairs</b>	
2	1,08
4	0,43
6	0,30
$8 \leq n \leq 40$	$0,23 \frac{8}{n}$

Figure 98 : Norme CEI-61000-3-2 pour les appareils de classe A

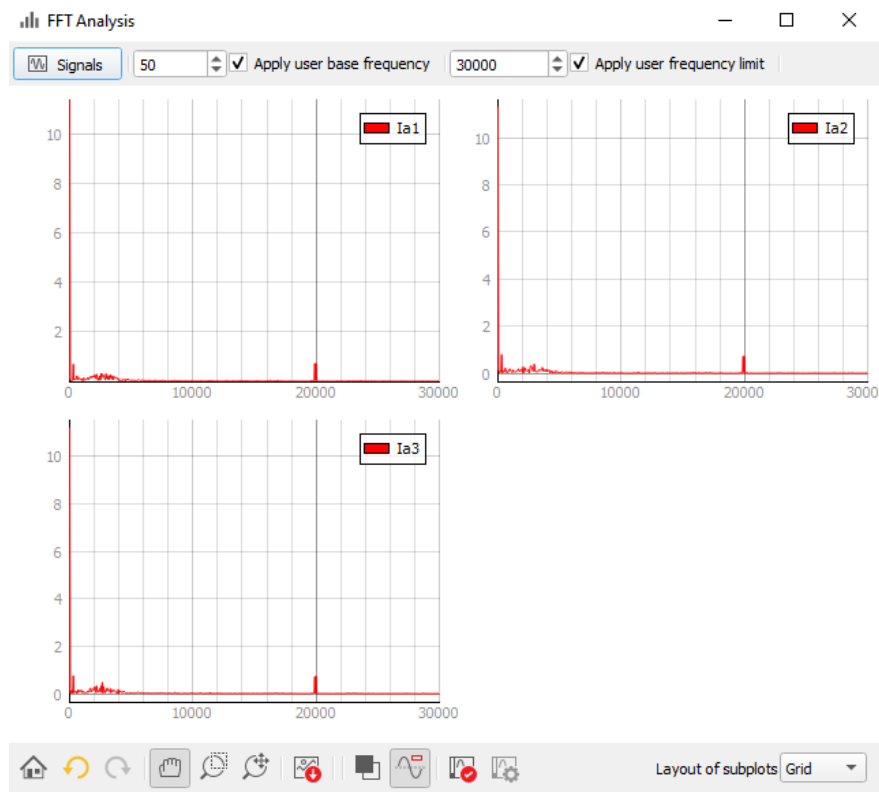


Figure 99 : FFT sur les 3 courants, 700VDC,  $I_d = 10A$  avec le réseau activé

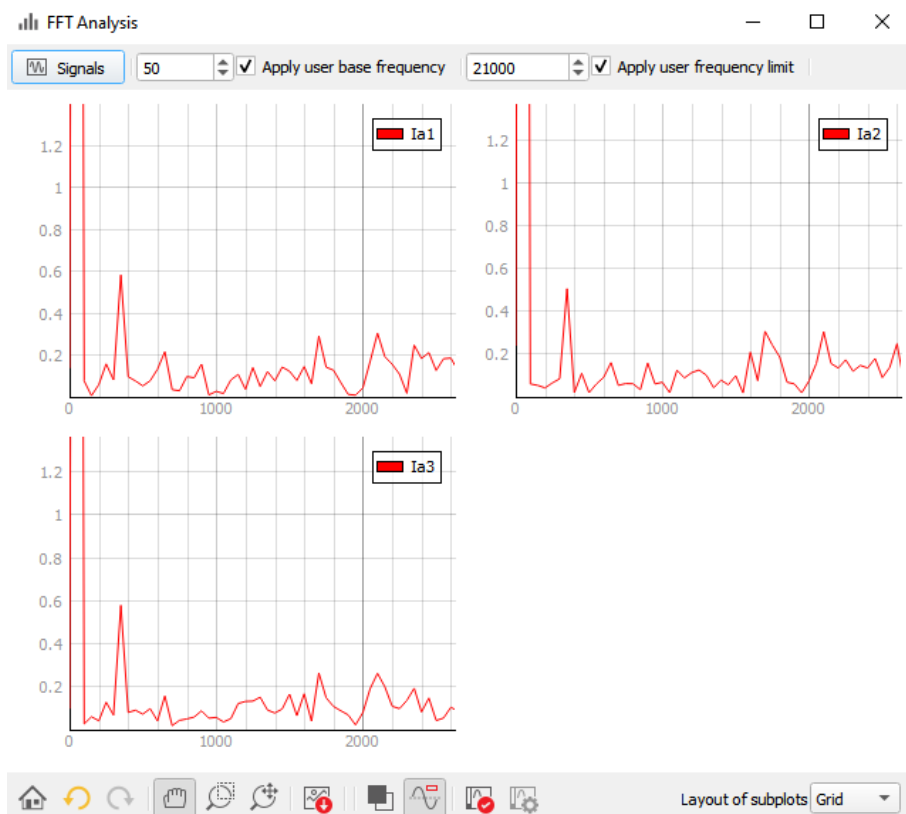


Figure 100 : FFT sur les 3 courants avec zoom 0 à 2kHz, 700VDC,  $I_d = 10A$  avec le réseau activé

Héritier Daniel

Sion, le 18.08.22

Les harmoniques dépassent les valeurs de la norme entre 1'700 et 4'000Hz et à la fréquence de commutation du système 20kHz. En ajoutant un filtre adapté au système pour bloquer les signaux de ces valeurs critiques celui-ci peut passer la norme.

Test avec un filtre passe-bas.

$R = 1 \Omega$

$C = 80\mu F$

$f_c = 2'000 \text{ Hz}$

Il reste uniquement des harmoniques à la fréquence de commutation (Figure 103)

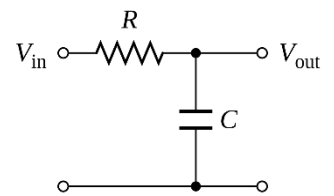


Figure 101 : Circuit filtre passe-bas

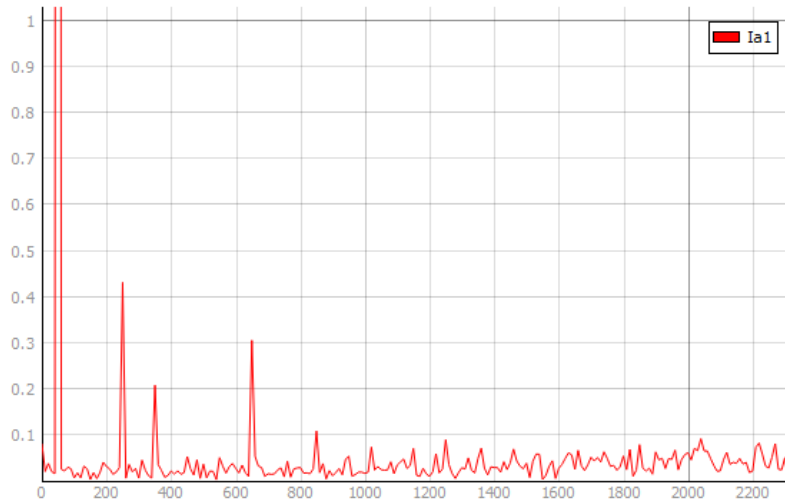


Figure 102 : Les fréquences qui posent problèmes à environ 2000Hz jusqu'à 20kHz sont atténuées

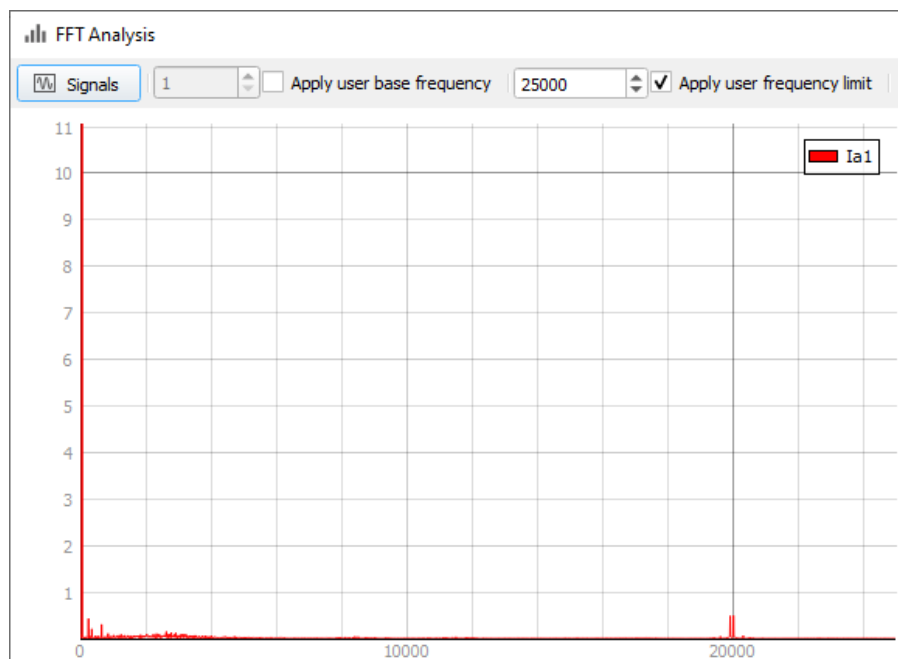


Figure 103 : Il reste uniquement les harmoniques à la fréquence de commutation de 20kHz

## 13 Test sur l'installation réelle charge RL

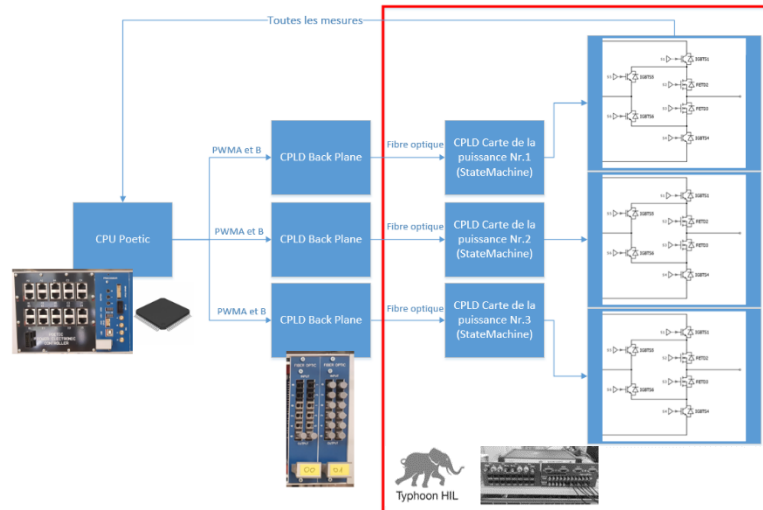


Figure 104 : Ce chapitre traite du système réel, il n'y a plus de simulation Typhoon

Les tests sur l'installation réelle commencent par des tests à basse puissances puis en augmentant progressivement. Il y a plusieurs tests à faire avant de lancer le système en régulation de courant en boucle fermée.

(Les tests sont répétitifs, il suffit de regarder le premier test ensuite on répète ces étapes pour différents niveaux de tension. Un résumé est présent à la fin au chapitre 13.6)

### 13.1 Test à une tension d'UDC = 60V

#### 13.1.1 Test de tension en sortie

Le tout premier test est de voir si le système commute correctement selon la commande du CPU Poetic. Pour faire le contrôle il suffit de mesurer la tension en sortie des trois phases. Il n'y a pas de courant en sortie de l'onduleur.

La commande sont des simples signaux PWM avec un duty cycle entre 0.0 et 1.0.

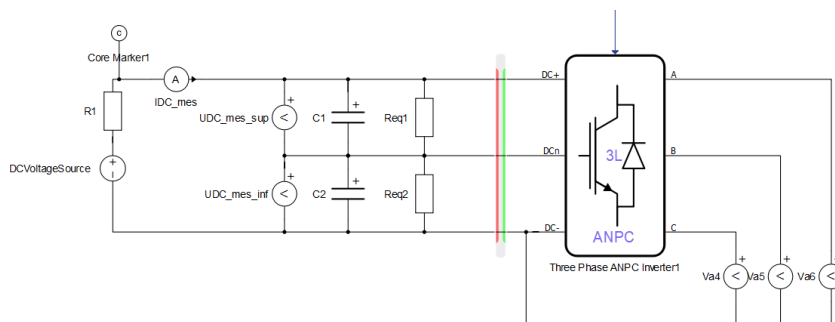


Figure 105 : Typhoon est uniquement utilisé pour montrer le schéma des tests ! Mesures de tensions en sortie de l'onduleur.

Le but est de voir si le duty cycle impacte la tension de sortie ainsi que les largeurs d'impulsions.

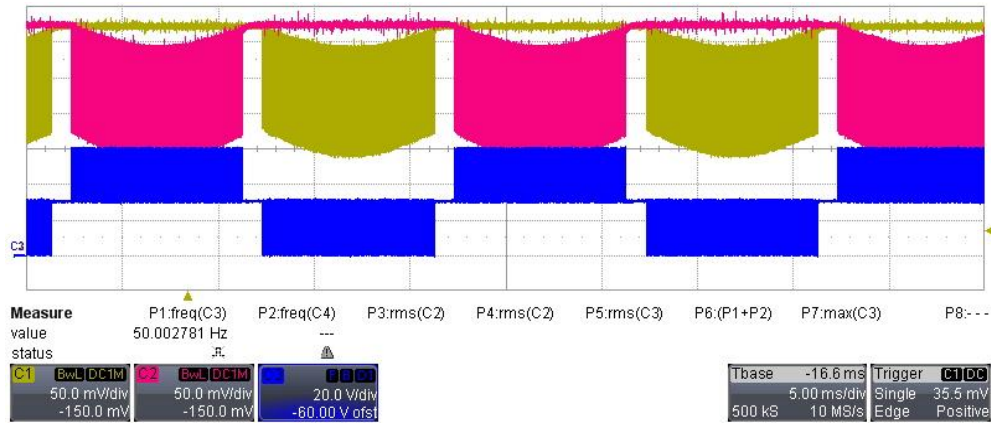


Figure 106 : Tension en sortie sur C3 avec signaux fibres optiques avec modulation. C1 = A, C2 = B (Mauvais branchement des fibres optiques, car c'est A qui doit moduler la partie positives. Inversion de HS et LS.)

Dans ce premier test il y a eu la découverte d'un problème de câblage des fibres optiques. Le branchement des sorties PWM A et B ne sont pas correctes. Il y a eu inversion sur le LS et HS. Afin de corriger ce problème il faut modifier les paramètres sur la CPLD de la carte de puissance. Après les modifications les tensions sont bonnes.

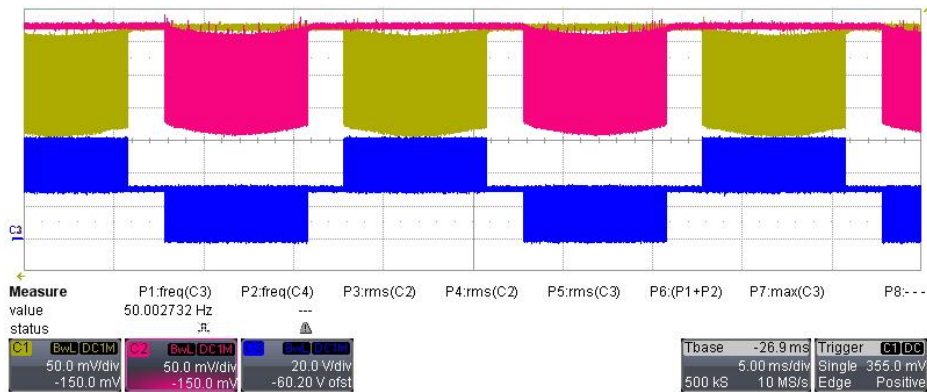


Figure 107 : Tension en sortie sur C3 avec signaux fibres optiques avec modulation. C1 = A, C2 = B Branchement correct. La fibre optique A, module la partie positive. HS et LS correct

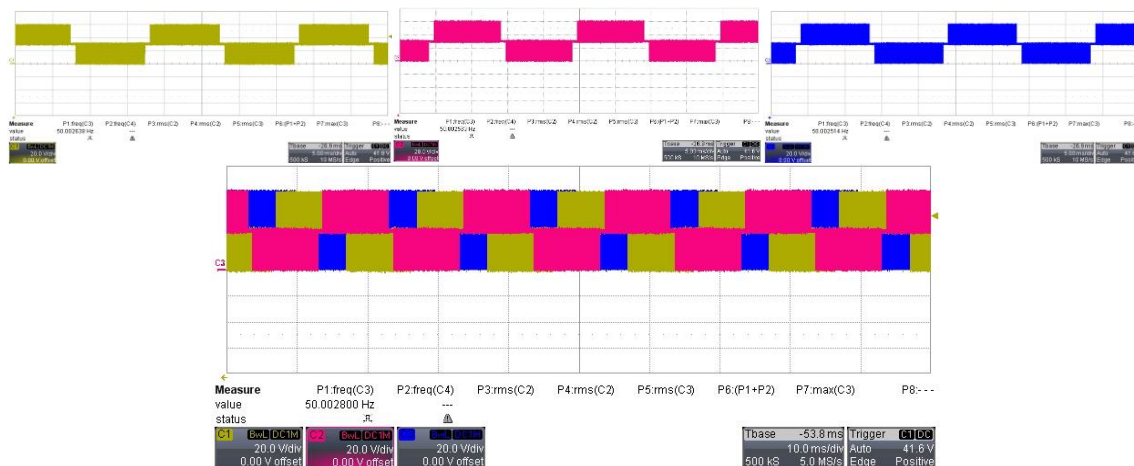


Figure 108 : Tensions de sorties du système individuelle puis combinaison des 3 phases

Modulation avec le duty cycle fait varier la tension de sortie :  
Duty Cycle faible (0.3) :



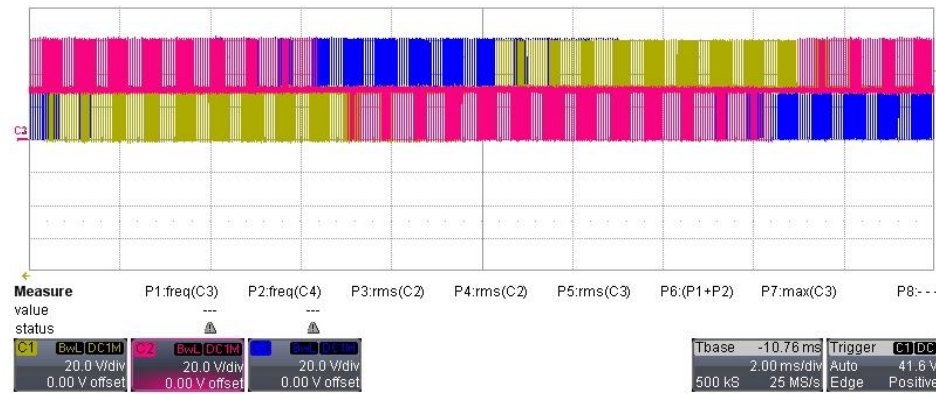


Figure 109 : Tension en sortie avec duty cycle à 0.3

La modulation avec un duty cycle au maximum est correcte.

Duty Cycle élevé (1.0) :

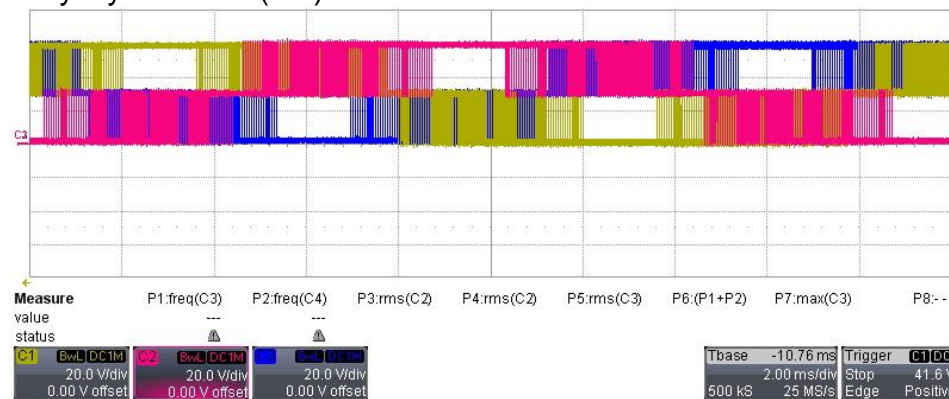


Figure 110 : Tension en sortie avec duty cycle à 1.0 (max)

Une fois que ces tests sont bons il est possible de brancher des charges sur les sorties de l'onduleur.

### 13.1.2 Test de courant avec charge

Pour les charges en sortie de système les valeurs choisies pour les résistances sont de 55 à 110 ohm variable et des inductances de 1mH.

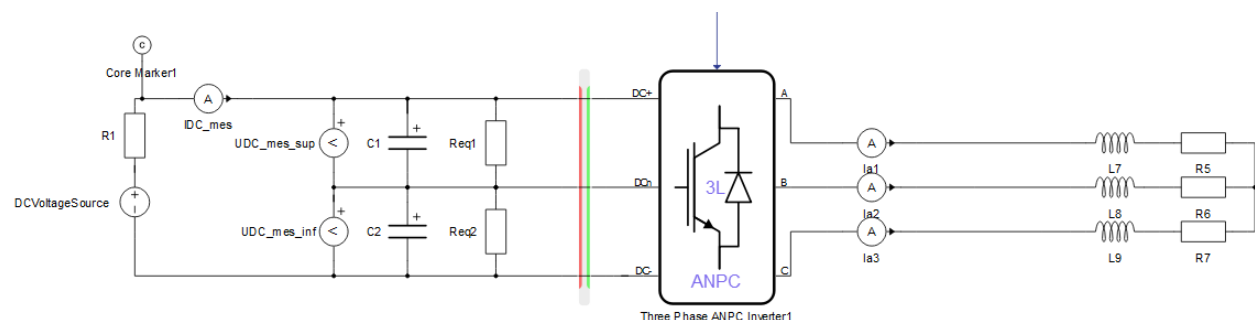


Figure 111 : Typhoon est uniquement utilisé pour montrer le schéma des tests ! Mesures de courants en sortie de l'onduleur avec charge RL

Pour augmenter le courant en sortie avec une tension DC de 60V il faut réduire la résistance. Car pour une certaine valeur de résistance on ne peut pas atteindre des courants élevés car la limite de modulation du système est atteinte.



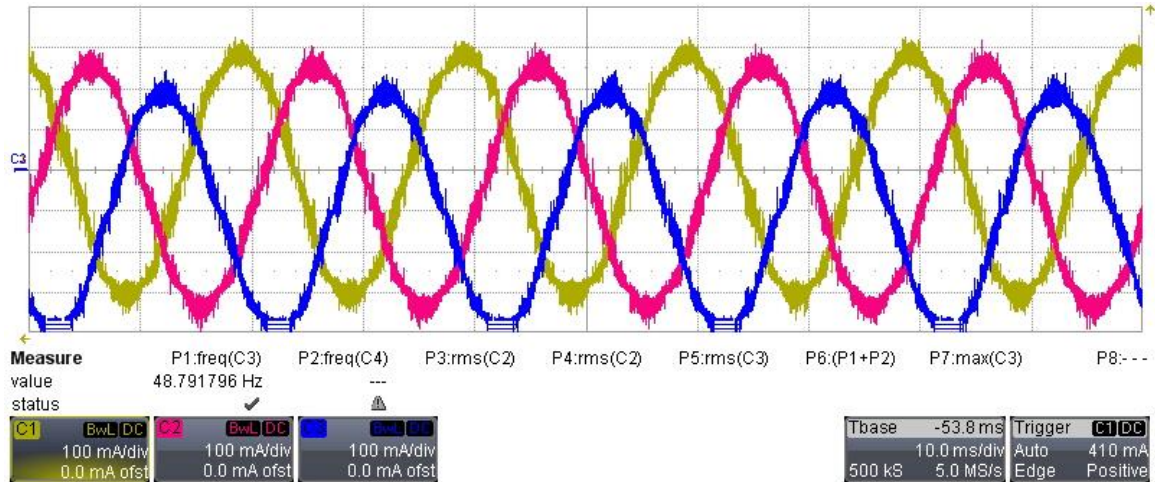


Figure 112 : Test à 60V et 110 ohm et  $I_d = 0.3A$ , (L'offset entre les mesures est dû au fait que les ampèremètres ne sont pas calibré, mais les amplitudes sont justes)

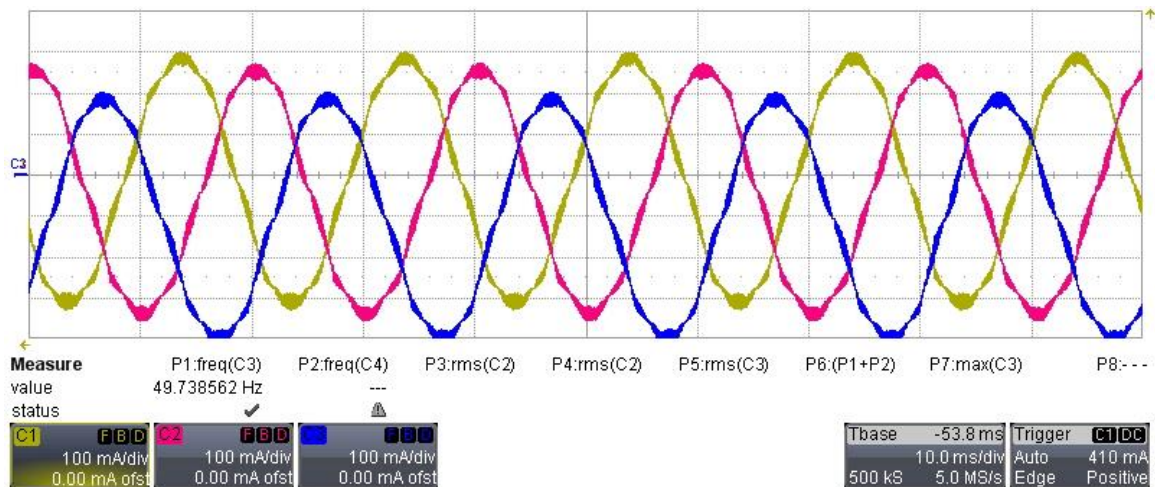


Figure 113 : Test à 60V avec 110 ohm et  $I_d = 0.3A$  avec réduction de bruit

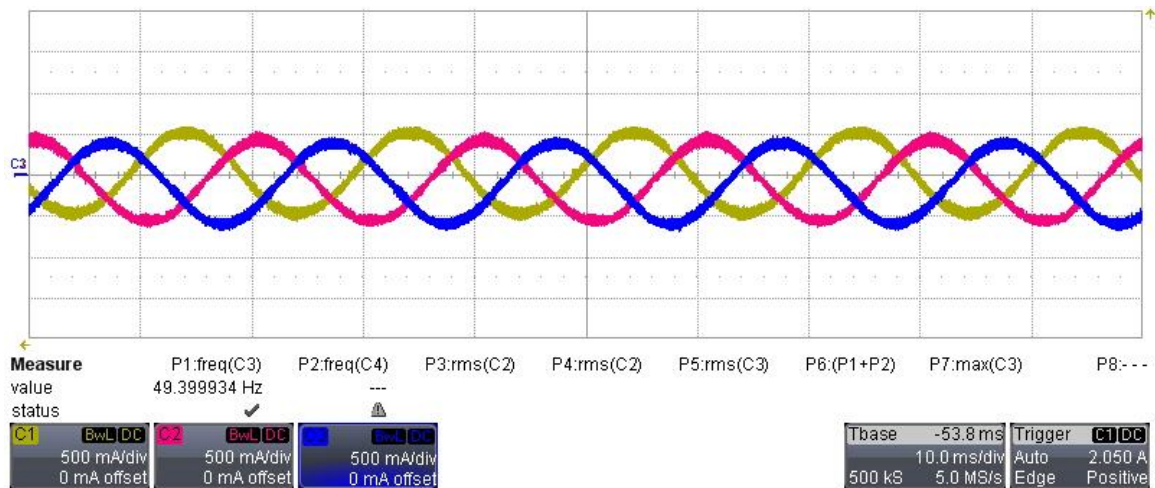


Figure 114 : Test à 60V avec 55 ohm et  $I_d = 0.5A$

Les valeurs de courants choisi aux figures précédentes avec  $I_d$  permettent de donner une consigne de courant, mais le but est d'utiliser  $I_d$  et  $I_q$  pour faire un décalage avec la tension du réseau pour produire des puissances actives et réactives.

### 13.1.3 Puissances actives et réactives

Pour contrôler le déphasage du courant en sortie il suffit de générer la tension de réseau avec Typhoon ce qui permet également de changer la fréquence du réseau simplement.

Pour une consigne de courant uniquement sur  $I_d$ , le courant devrait parfaitement être en phase avec la tension de réseau. Comparaison entre C1 et C4.

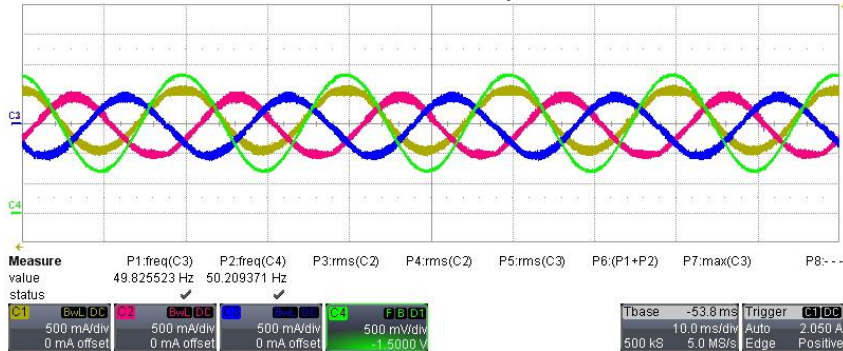


Figure 115 : Test à 60V avec 55 ohm et  $I_d = 0.5A$ , C4 est la tension de réseau, C1 est le courant de la phase 1 et est en phase avec C4

Pour une consigne de courant uniquement sur  $I_q$ , le courant devrait être décalé de 90 degrés avec la tension de réseau. Comparaison entre C1 et C4.

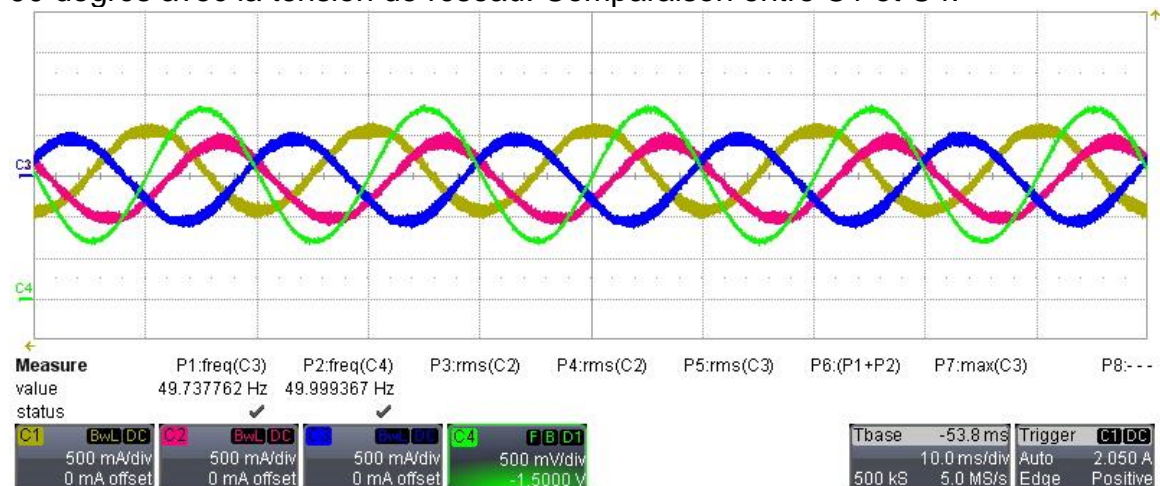


Figure 116 : Test à 60V avec 55 ohm et  $I_q = 0.5A$ , C4 est la tension de réseau, C1 est le courant de la phase 1 et est décalé de 90 degrés avec C4

Des courants  $I_d$  et  $I_q$  identiques doivent fournir un courant de sortie décalé de 45 degrés avec la tension du réseau. Comparaison entre C1 et C4.

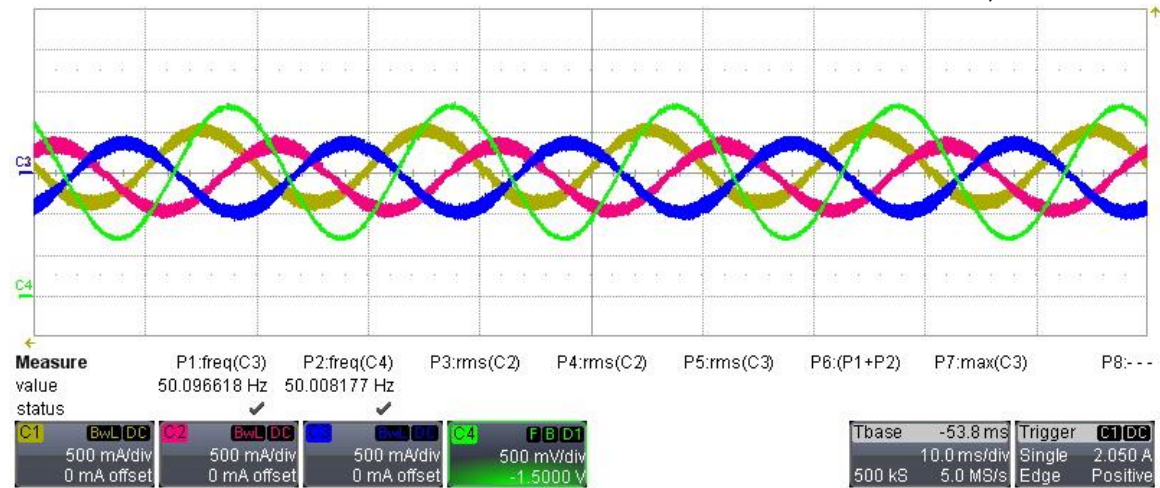


Figure 117 : Test à 60V avec 55 ohm et  $I_d$  et  $I_q = 0.3A$ , C4 est la tension de réseau, C1 est le courant de la phase 1 et est décalé de 45 degrés avec C4

### 13.1.4 Test de fréquence

La sortie de l'onduleur doit également s'adapter à différentes fréquences de réseau.

Si le réseau est à 40Hz il n'est pas possible d'injecter du 50Hz. Il faut pouvoir modifier la vitesse de notre système.

En changeant la fréquence du réseau sur Typhoon on peut faire ce test.

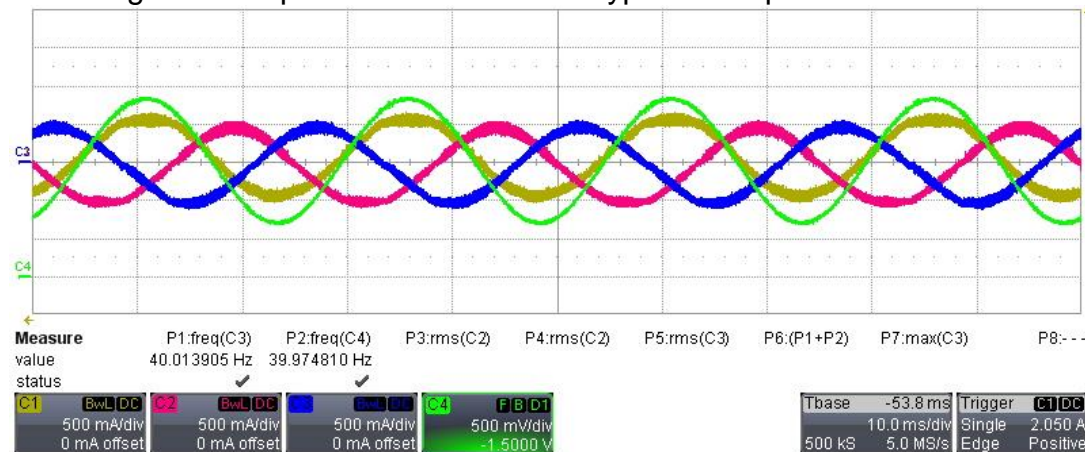


Figure 118 : Test à 60V avec 55 ohm et  $I_d = 0.5A$ , C4 est la tension de réseau à 40Hz. C1 est le courant de la phase 1 et il est en phase avec le réseau avec la bonne fréquence de 40Hz



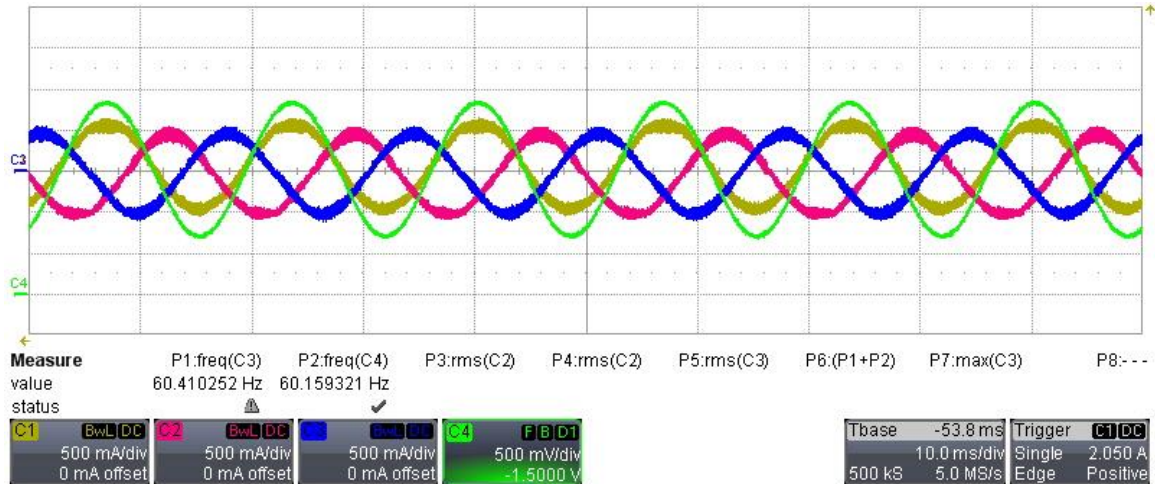


Figure 119 : Test à 60V avec 55 ohm et  $I_d = 0.5A$ , C4 est la tension de réseau à 60Hz. C1 est le courant de la phase 1 et il est en phase avec le réseau avec la bonne fréquence de 60Hz

## 13.2 Test à une tension d'UDC = 120V

### 13.2.1 Test de courant avec charge

Pour les charges en sortie de système nous avons choisi de mettre des résistances de 55 ohm et des inductances de 1mH.

### 13.2.2 Puissances actives et réactives

Pour contrôler le déphasage du courant en sortie il suffit de générer la tension de réseau avec Typhoon ce qui permet également de changer la fréquence du réseau simplement.

Pour une consigne de courant uniquement sur  $I_d$ , le courant devrait parfaitement être en phase avec la tension de réseau.

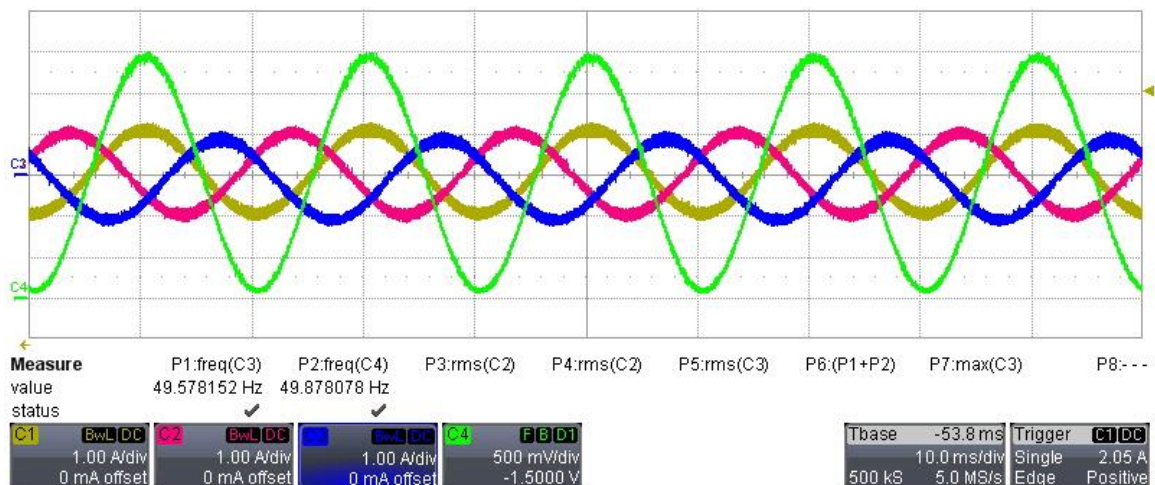


Figure 120: Test à 120V avec 55 ohm et  $I_d = 1.0A$ , C4 est la tension de réseau, C1 est le courant de la phase 1 et est en phase avec C4

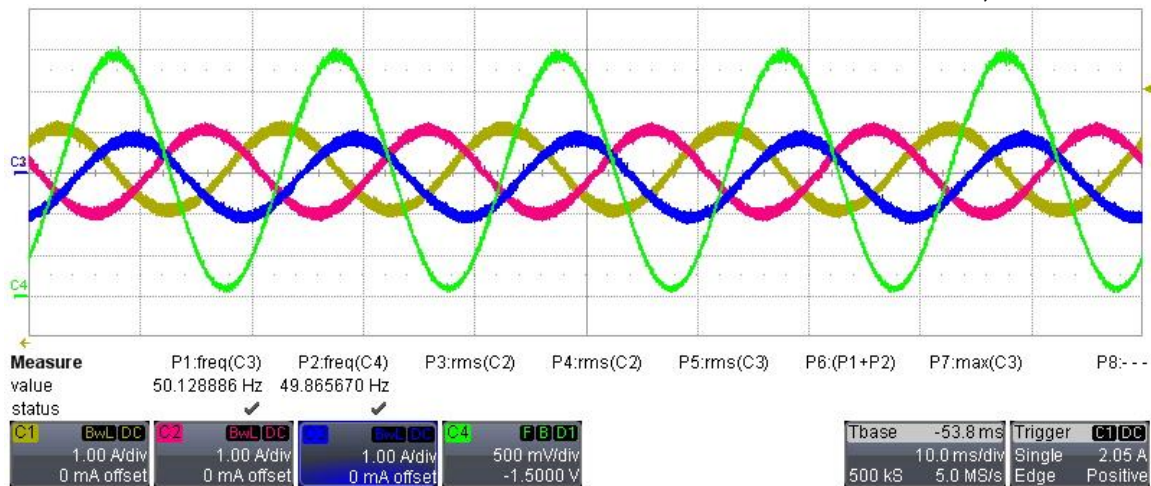


Figure 121 : Test à 120V avec 55 ohm et  $I_q = 1.0A$ , C4 est la tension de réseau, C1 est le courant de la phase 1 et est décalé de 90 degrés avec C4

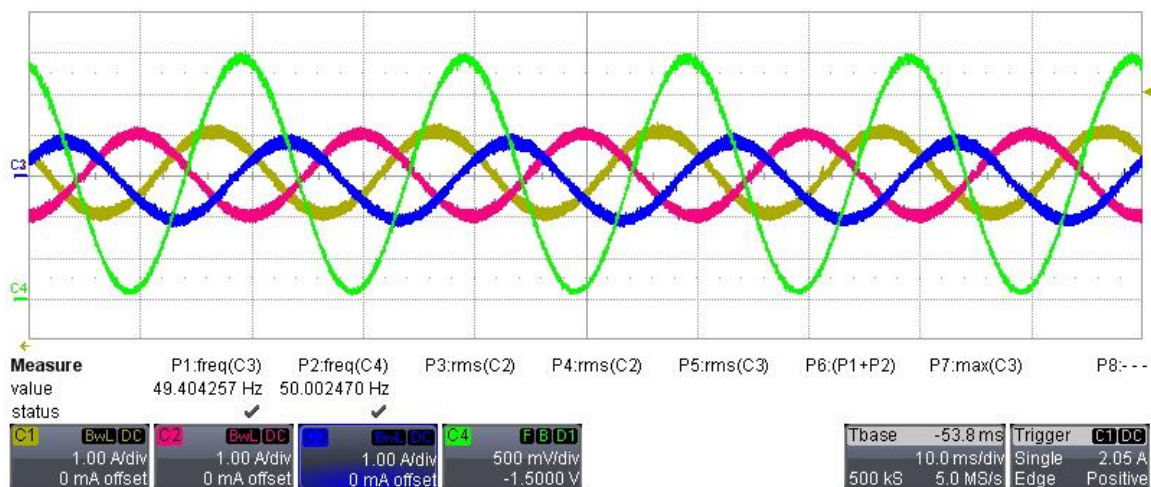


Figure 122 : Test à 120V avec 55 ohm et  $I_d$  et  $I_q = 0.7A$ , C4 est la tension de réseau, C1 est le courant de la phase 1 et est décalé de 45 degrés avec C4

### 13.2.3 Test de fréquence

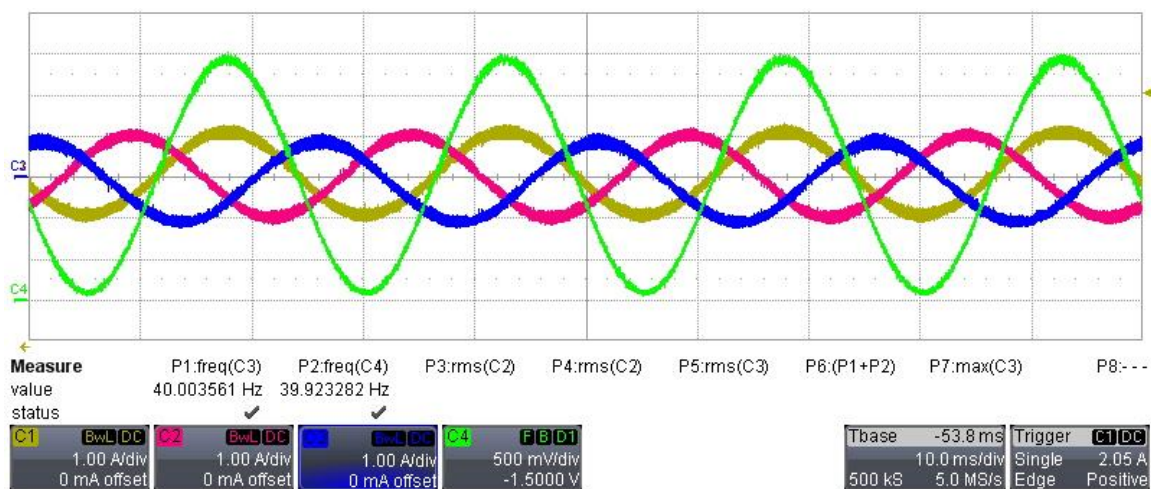


Figure 123 : Test à 120V avec 55 ohm et  $I_d = 1.0A$ , C4 est la tension de réseau à 40Hz. C1 est le courant de la phase 1 et il est en phase avec le réseau avec la bonne fréquence de 40Hz



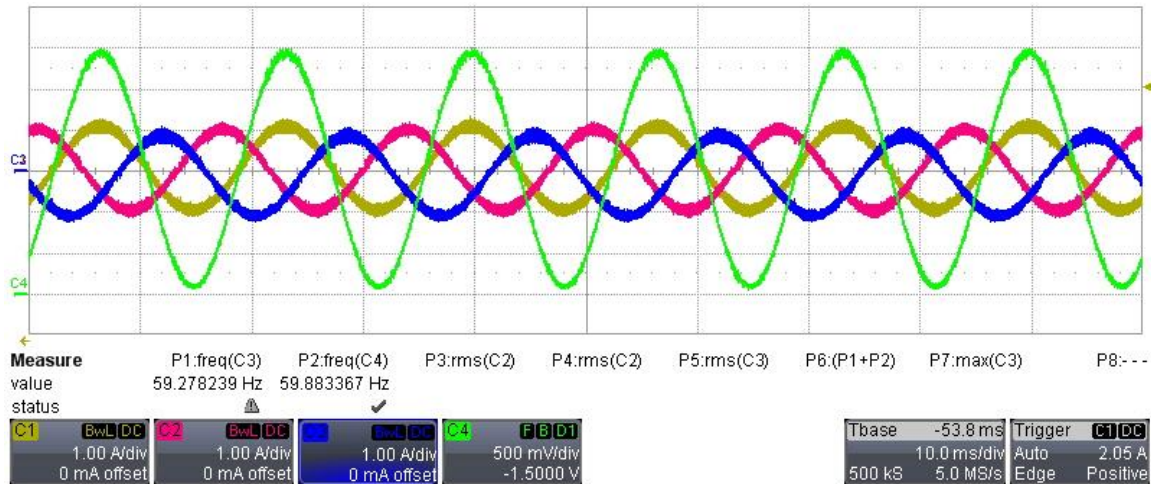


Figure 124 : Test à 120V avec 55 ohm et  $I_d = 1.0A$ , C4 est la tension de réseau à 60Hz. C1 est le courant de la phase 1 et il est en phase avec le réseau avec la bonne fréquence de 60Hz

### 13.3 Test à une tension d'UDC = 200V

Résistance, 82.5ohm. Inductance 1mH

#### 13.3.1 Test de courant en charge

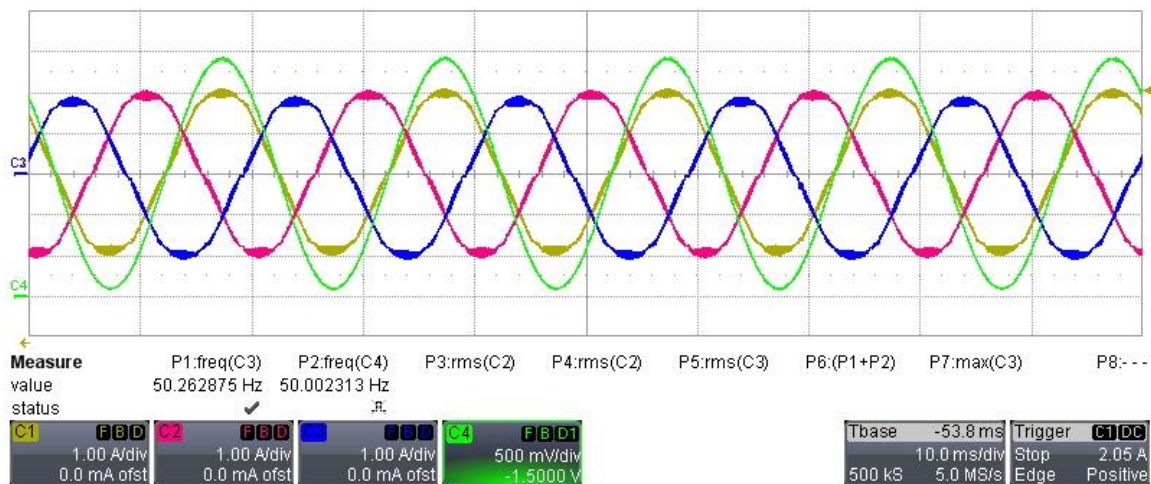


Figure 125 : Test à 200V avec 55 ohm et  $I_d = 2.0A$ , C4 est la tension de réseau, C1 est le courant de la phase 1 et est en phase avec C4

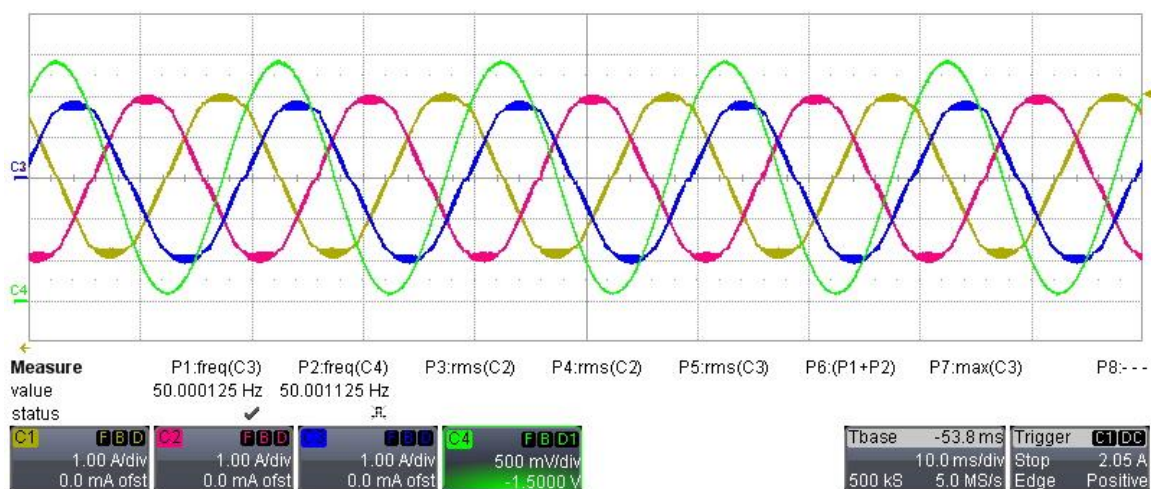


Figure 126 Test à 200V avec 55 ohm et  $I_q = 2.0A$ , C4 est la tension de réseau, C1 est le courant de la phase 1 et est décalé de 90 degrés avec C4

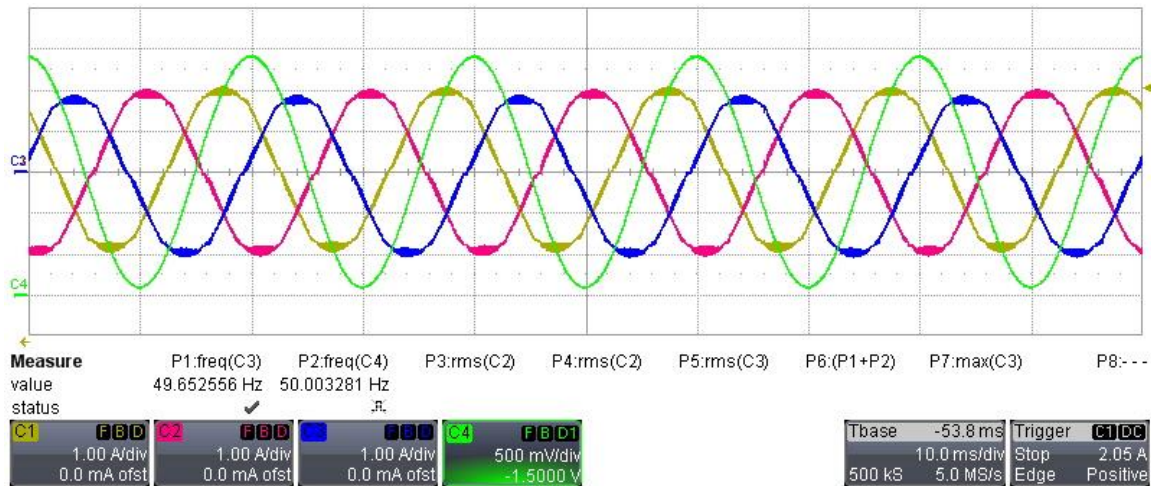


Figure 127 : Test à 200V avec 55 ohm et  $I_d$  et  $I_q = 1.41A$ , C4 est la tension de réseau, C1 est le courant de la phase 1 et est décalé de 45 degrés avec C4

### 13.3.2 Test de fréquence

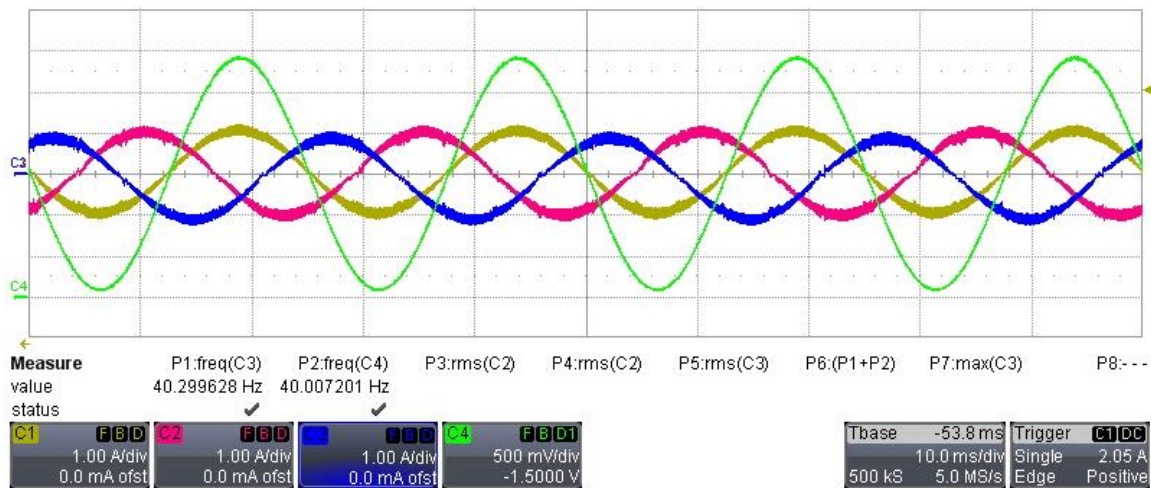


Figure 128 : Test à 200V avec 82.5 ohm et  $I_d = 1.0A$ , C4 est la tension de réseau à 40Hz. C1 est le courant de la phase 1 et il est en phase avec le réseau avec la bonne fréquence de 40Hz

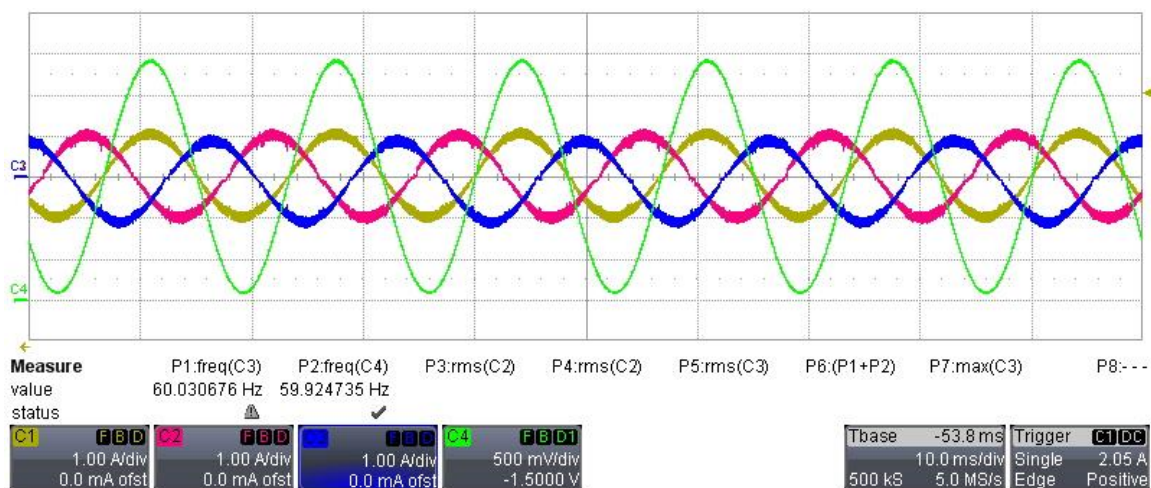


Figure 129 : Test à 200V avec 82.5 ohm et  $I_d = 1.0A$ , C4 est la tension de réseau à 60Hz. C1 est le courant de la phase 1 et il est en phase avec le réseau avec la bonne fréquence de 60Hz



### 13.4 Test d'équilibre des tensions DC

Jusqu'à ici les tests se sont faits avec deux alimentations. Chaque alimentation se branche sur le bus DC supérieur et inférieur. Mais pour continuer à augmenter la puissance il faut avoir une alimentation plus grande et utiliser uniquement une seule. Les 200 V vont se brancher sur les condensateur supérieur et inférieur en même temps et ce test permet de voir si l'équilibre entre la tension supérieur et inférieur est bon. Exemple : Pour un bus DC de 200V il faut obtenir une tension de 100V pour le bus DC supérieur et 100V pour le bus DC inférieur.

Résultat :

200V	DC Bus sup	DC Bus inf
Off	99.9 V	99.9 V
$I_d = 0.0, I_q = 0.0$	109.9 V (instable)	89.9 V (instable)
$I_d = 1.0, I_q = 0.0$	99.0 V	100.8 V
$I_d = 0.0, I_q = 1.0$	98.9 V	100.9 V
$I_d = 0.5, I_q = 0.5$	99.5 V	100.3 V

Tableau 10 : Tableau des mesures DC sur les deux condensateurs

En fonctionnement avec des consignes de courants d' $I_d$  et d' $I_q$  le système est plutôt équilibré et stable (en augmentant le courant il ne varie quasiment pas). Mais sans la tension de réseau le système devient instable si aucune consigne de courants  $I_d$  et  $I_q$  n'est défini.

En ajoutant l'équilibre des tensions ce problème n'existe plus. (Chapitre 12.6)



Figure 130 : Equilibrage entre Bus DC supérieur et inférieur (gauche = sup, droite = inf) lorsque l'installation est à l'arrêt



Figure 131 : Equilibrage entre Bus DC supérieur et inférieur (gauche = sup, droite = inf) lorsque l'installation est à des consignes de courants  $I_d$  et  $I_q = 0A$  (Tensions instables, mais avec l'équilibre des tensions ce problème n'existe plus)

### 13.5 Test à une tension de 600V

Avec des tensions plus élevées il y a des problèmes de commutation et bruits. (Plus d'explications au chapitre 15)

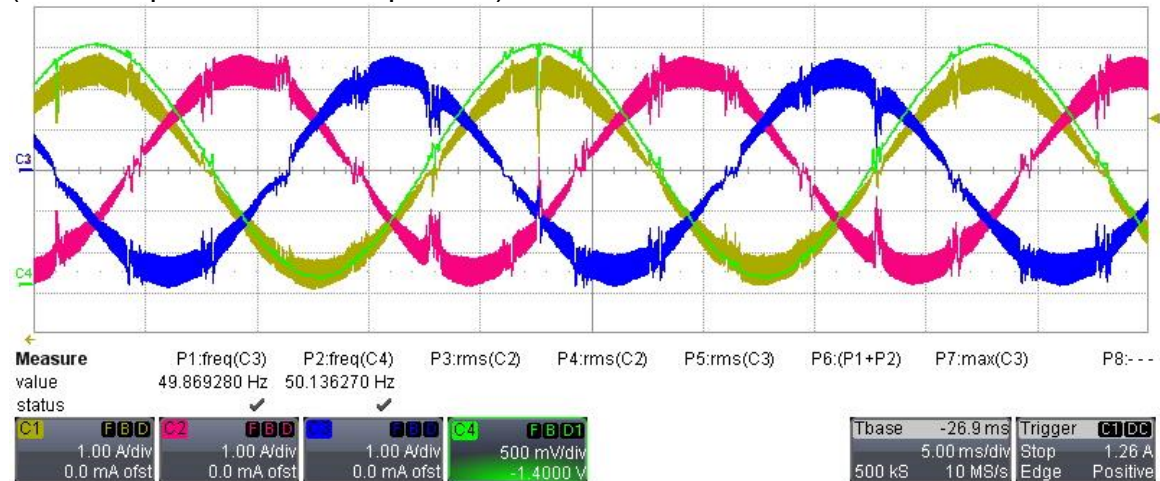


Figure 132 : Test à 600V avec 100 ohm et  $I_d = 2.5A$ , C4 est la tension de réseau, C1 est le courant de la phase 1 et est en phase avec C4

### 13.6 Résumé des tests sur l'installation réelle avec des charges RL

Pour tous les niveaux de tensions il y a une mesure avec  $I_d$  seul,  $I_q$  seul et combinaison d' $I_d$  et  $I_q$  ainsi qu'un changement de fréquence. Ceci permet de tester la plupart des cas.

Une mesure des tensions DC de la partie supérieur et inférieur est toujours faite pour assurer que l'installation ne soit pas en déséquilibre.

Pour les tests de 60 et 120 V le déséquilibre n'est pas possible car le branchement se fait avec deux alimentation. Pour les tests avec une tension plus importante l'équilibre est fait par les condensateurs. L'équilibre est bon.

Le système avec deux alimentations permet de garder un équilibre en tout temps sur les condensateurs. Donc pour le futur si deux alimentations sont disponibles c'est un meilleur choix pour faire des tests en gardant l'équilibre.

## 14 Expérimentation avec réseau industriel, mesure et validation.

Les tests sur l'installation avec le réseau industriel ne sont pas satisfaisant. Les courants ne ressemblent pas du tout à des sinus. Après plusieurs modifications du code (changement des gains de régulateurs) et équilibre des tensions DC rien n'aide pour améliorer les résultats. La fréquence des courants est correcte à 50Hz et le déphase également (C1 en phase avec C4, car consigne sur  $I_d$  uniquement). L'amplitude du signal est d'environ 2.0A mais ce n'est pas du tout régulier sur la forme du sinus. Le chapitre 15 émet des hypothèse pour ces problèmes.

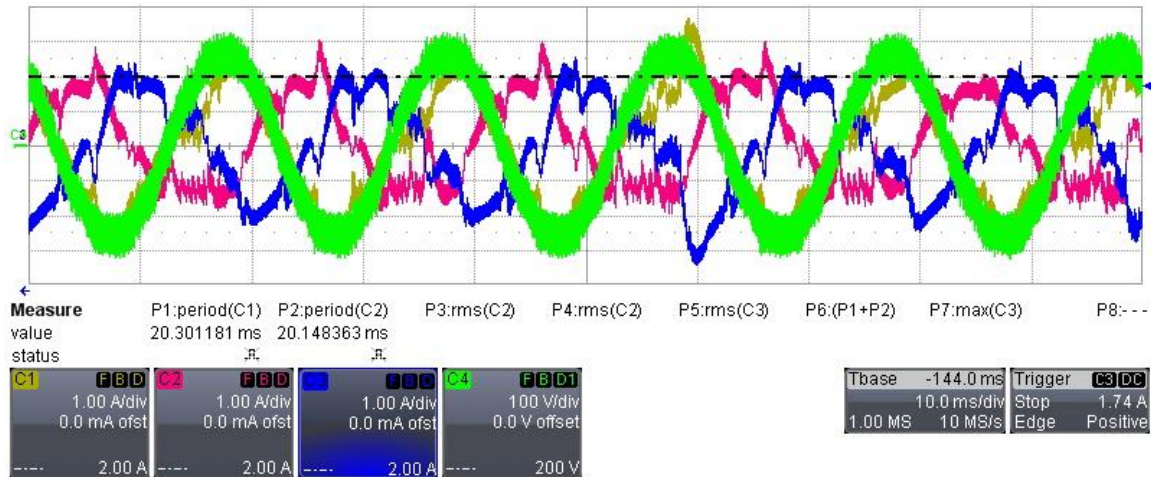


Figure 133 : C1 à C3 = Courants des trois phases, C4 = Tension de réseau phase 1

## 15 Problèmes sur l'installation réelle avec le réseau connecté

Selon les tests fait en simulation avec Typhoon ou sur l'installation réelle avec une charge RL le système fonctionne très bien. Mais pour les tests avec l'installation branchée au réseau il y a des problèmes de fonctionnement. En augmentant la consigne de courant le système augmente sa puissance mais les courants de sortie ne ressemblent pas du tout à des sinus.

C1 = Courant phase 1

C2 = Courant phase 2

C3 = Courant phase 3

C4 = Tension de réseau U1

Le déphasage des phases semble correct, mais les valeurs ne sont pas du tout régulières.

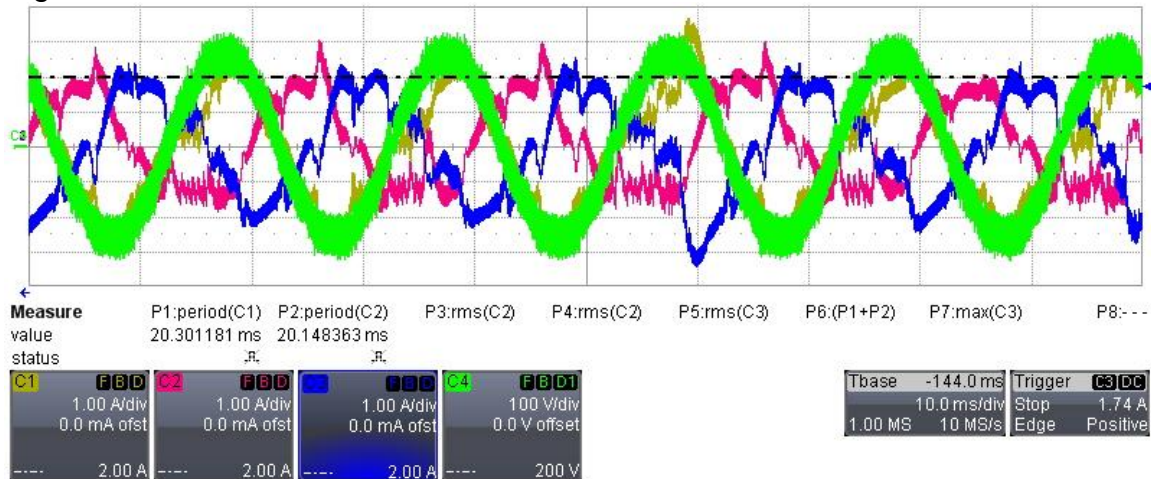


Figure 134 : Problème avec les courants en sortie de l'onduleur C1 à C3 = Courants en sortie de l'onduleur, C4 = Tension de réseau de la phase 1

En essayant de trouver où les problèmes se trouvent il y a eu quelques découvertes.



La tension de commande de la carte de puissance ne reste pas au 3.3 V, mais oscille à une fréquence de 1MHz et fait commuter les semi-conducteurs et ainsi la tension de sortie.

C1 = 3.3 V Tension de commande de la carte de puissance (Avec condensateur pour tenter de stabiliser la tension)

C4 = Tension de sortie de l'onduleur (Mesure Uphase1out-0V)

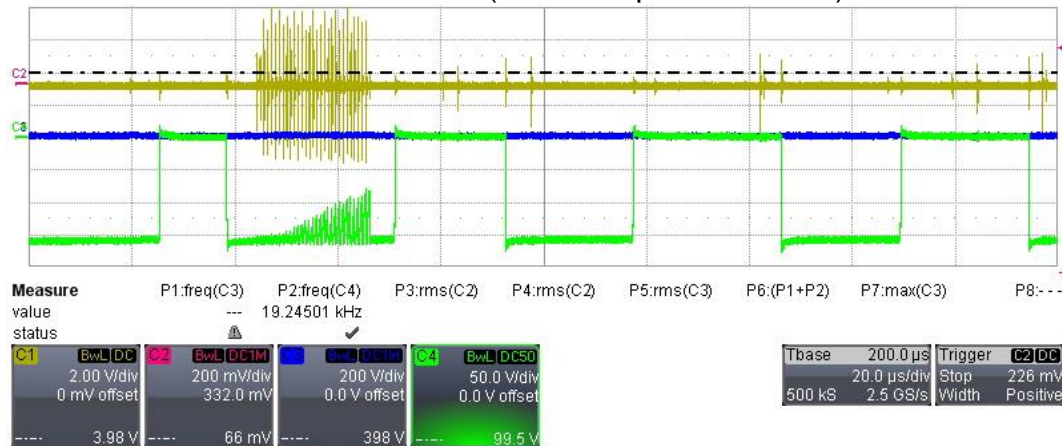


Figure 135 : C1 = Tension de 3.3V de la commande des cartes de puissance, C4 = Tension de sortie de l'onduleur(Mesure entre Uphase1out-0V)

C1 = Tension de commande de 3.3V de la carte de puissance

C2 = Tension sur la commande de la fibre optique

C3 = Tension de sortie de l'onduleur sur la phase 2 (Mesure Uphase2out-0V)

C4 = Tension de sortie de l'onduleur sur la phase 1 (Mesure Uphase1out-0V)

Le fonctionnement est ok mais de temps en temps il y a ces oscillations d'environ 1MHz qui perturbent tout le système.

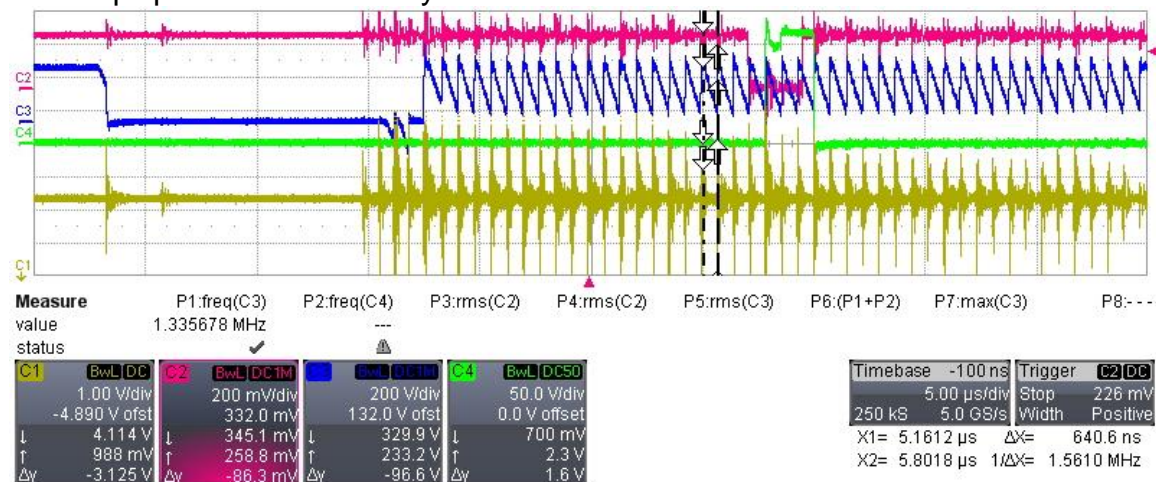


Figure 136 : C1 = Tension de 3.3 V de commande de la carte de puissance, C2 = Tension sur la commande avec la fibre optique, C3 = Tension de sortie de l'onduleur sur la phase 2 (Mesure Uphase2out-0V), C4 = Tension de sortie de l'onduleur sur la phase 1 (Mesure Uphase1out-0V)

Le module qui fournit le 3.3V n'est pas saturé car il peut fournir 1.5 A mais il consomme uniquement 700mA.

Les mesures sont également très bruitées et arrivent faux dans le CPU qui ensuite fait des mauvais calculs pour la régulation.

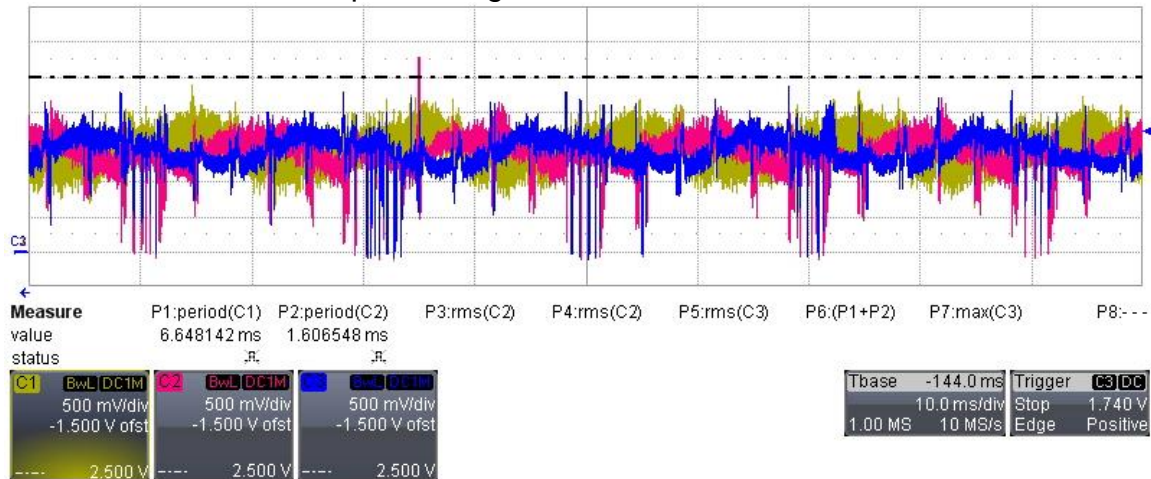


Figure 137 : Les valeurs que le CPU Poetic a reçu pour les courants, C1 à C3 = I1, I2 et I3

L'hypothèse est que c'est un problème d'EMC dans le système qui perturbe les mesures et les tensions de commande qui fait que le système ne peut pas fonctionner correctement.

## 16 Remarque

Les branchements de l'installation sont disponibles en annexe (19.7).

Le mode d'emploi pour le système avec Typhoon ou le système réel est en annexe (19.13)

Les fichiers informatiques, Simulation Plecs, Simulation Typhoon, Code Poetic et fichiers Matlab sont enregistrés sur le gitlab.hevs ANPC (Groupe privé).

## 17 Conclusion

Le début du projet a permis de comprendre le fonctionnement de la topologie ANPC en effectuant des recherches [1][2][4]. Ceci a permis de créer une structure complexe qui pilote le système avec la state machine des cartes de puissance. Suivi par un développement de cette state machine pour y ajouter des états de protection en cas de défaut des semi-conducteurs. Ces connaissances ont permis de faire des simulations du système sur PLECS et obtenir des résultats concluants avec toutes les régulations déjà opérationnelles.

L'étape suivante est le code pour la carte CPU de Poetic, qui correspond au cerveau de l'installation. Plusieurs tests ont été faits pour avoir le fonctionnement voulu en sortie sur les fibres optiques. Ensuite la simulation Typhoon HIL est faite qui permet de remplacer les cartes de puissance par une simulation Hardware in the Loop. Avec Poetic et Typhoon en parallèle, tous les tests de fonctionnement ont pu être fait. Ceci a permis de définir les gains des régulateurs de l'installation ou les tensions DC minimum pour ne pas être saturé. Ces simulations fonctionnent très bien et ont permis de confirmer le bon fonctionnement des régulations de courants en sorties selon les consignes de l'utilisateur ou le déphasage pour produire de la puissance active ou réactive. Les deux méthodes de modulation PWM et SVPWM fonctionnent.

Lorsque les tests avec Typhoon sont terminés il est possible de passer avec les cartes de puissances réels. Les tests sont concluant pour une charge RL en sortie.

Cependant il y a eu des problèmes pour le fonctionnement en branchant le système sur le réseau. Du bruit important perturbe les signaux de mesures mais également les tensions de commande du système. Ceci génère des problèmes sur les courants de sorties. Les mesures sont bruitées ce qui gêne la régulation qui fait de mauvais calculs et perturbe encore plus les sorties. L'hypothèse est que ces problèmes apparaissent car il y a des problèmes d'EMC. Cette dernière partie n'est ainsi pas opérationnelle pour le moment.

## 18 Références

- [1] Jonathan Dodge, "3L-ANPC vs. 3L-NPC Inverters", Application UnitedSiC\_AN0023 2020
- [2] Y. Liu, C. Hu, Q. Chen, D. Ma and L. Zhao, "Neutral-point potential balancing strategy for three-level ANPC converter base on discontinuous PWM," 2017
- [3] Gabriele Pescatore, "Nouvelles technologies adaptées à l'enseignement des entraînements électriques", 2020
- [4] L.Halabi, I.Alsofyani, K-B. Lee " An improved Space Vector Modulation for Short-Circuit Fault-Tolerance in Multi-Level Hybrid ANPC", Ajou University Korea 2021

## 19 Annexes

- Etat dangereux dans le fonctionnement
- Visio State machine PWM4 basique
- Visio State machine PWM4 avec protection de défaut
- Visio Régulation Système DQ
- Courbe de simulation des régulations PLECS
- Calcul pour la méthode Space Vector Modulation
- Vecteurs à choisir selon le cas
- Matlab Space Vector Modulation comparaison des deux options
- Cablage/Branchement de l'installation avec Typhoon ou système réel
- Visio State Machine de l'installation
- Code Poetic
- Code Space Vector modulation





## 19.2 Protection de la state machine



### ARROW CASE 1

There is a condition to change from any state to another

### AND

Once you enter a state, start a timer. Once the timer is over you can leave the state if the command is activated



### ARROW CASE 2

Arrows marked as 'Direct no waiting needed' are dangerous states (short-circuit) and we need to change the state as fast as we can. No need to wait for timer

### Timings for this project

Ton, Toff → Sw1, Sw4, Sw5, Sw6 (IGBT)  
TonMid, ToffMid → Sw2, Sw3 (Mosfet)

Ton = 90 ns  
Toff = 520 ns  
TonMid = 30 ns  
ToffMid = 50 ns

In Hybrid mode we use the longer timings because we never switch Mosfets alone

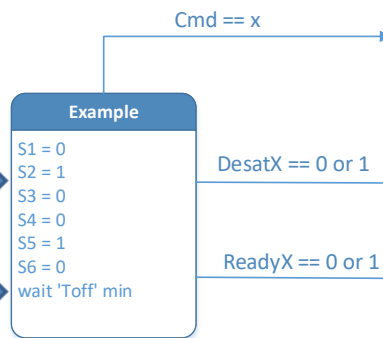
## Example and explanations

Cmd == x is to change state while the system is working without issues

Cmd can be these values :  
0, 1, 2 and 3

6 components :  
**Sx** = 0 or 1  
If the component should be turned ON or OFF

**Wait** is for the minimum time we need to wait in 'normal working conditions'



There are 6 signals of 'desat' and 'ready' for each component that protects the circuit:

Desat signals are if a problem is detected. Can be a short-circuit, we need to be careful and take special steps (0 = ok, 1 = issue detected)

Ready signals are if there is an undervoltage or the temperature is outside of the acceptable range (0 = not ready, 1 = ready to start)

Desat and Ready signals have priority and don't need to wait if activated

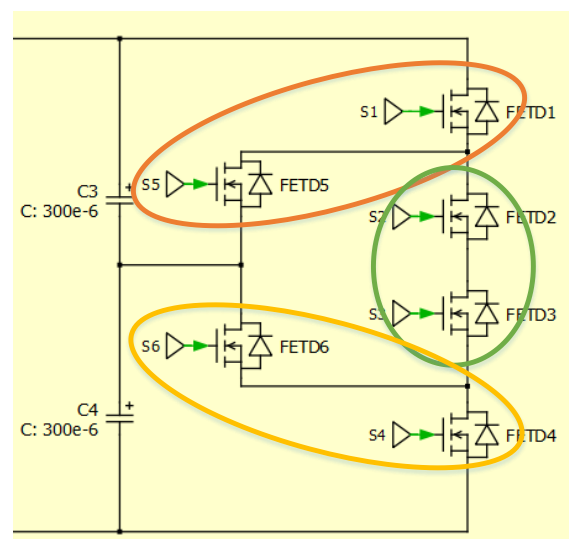
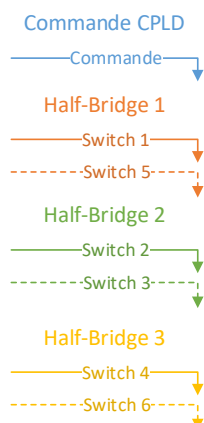


Figure 139 : Explications pour comprendre le fonctionnement de la state machine avec les protections, (Figure 15)



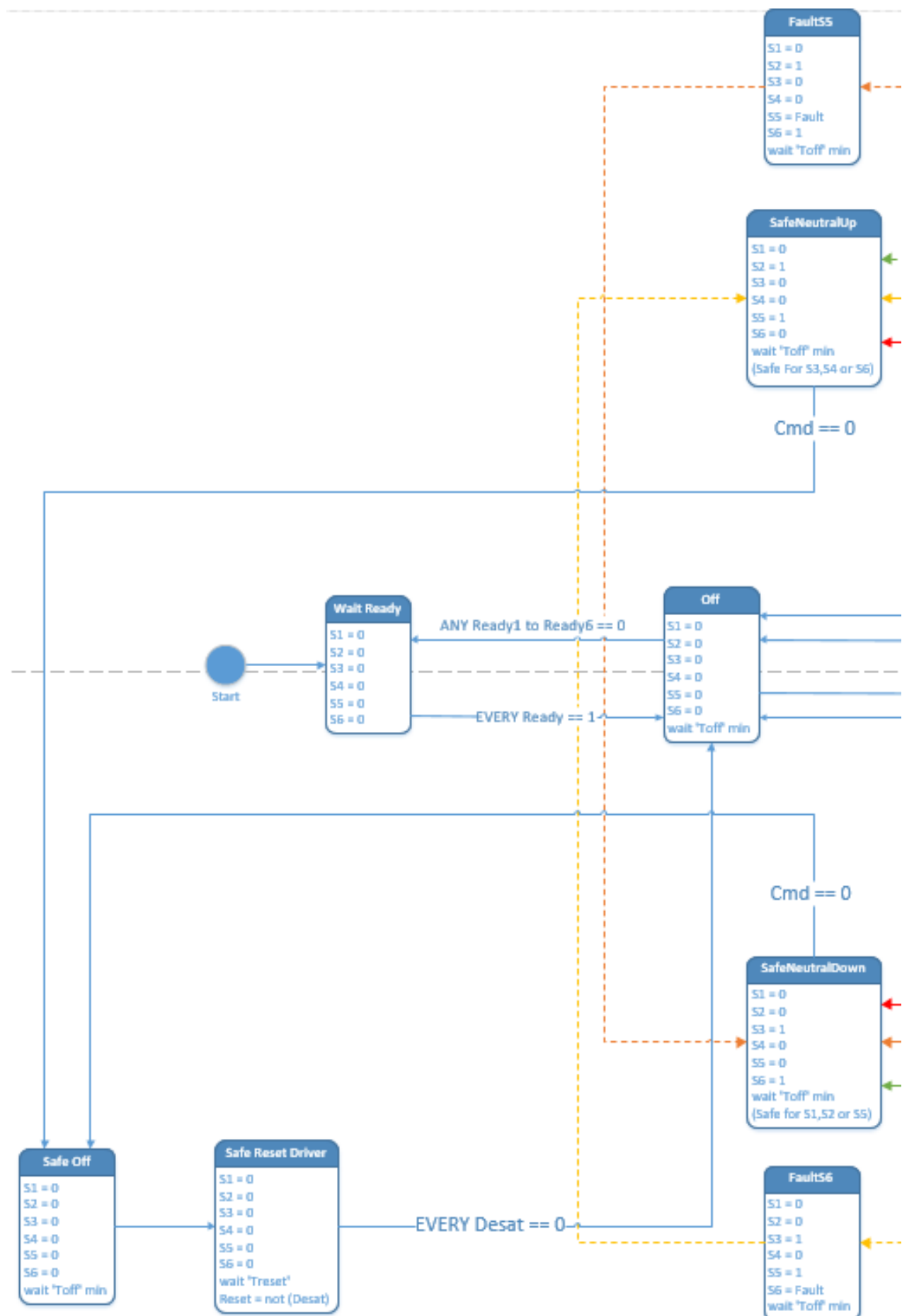


Figure 140 : Machine d'état PWM4 avec protection desat et ready (partie 1 gauche)(Figure 16)



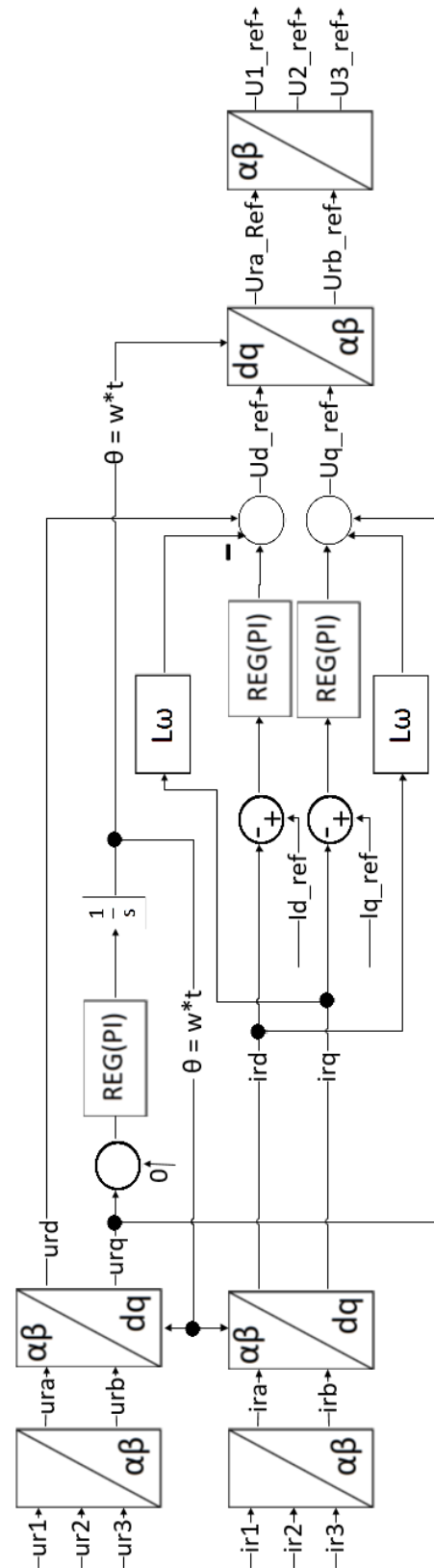


Figure 142 : Schéma de régulation (Figure 30)

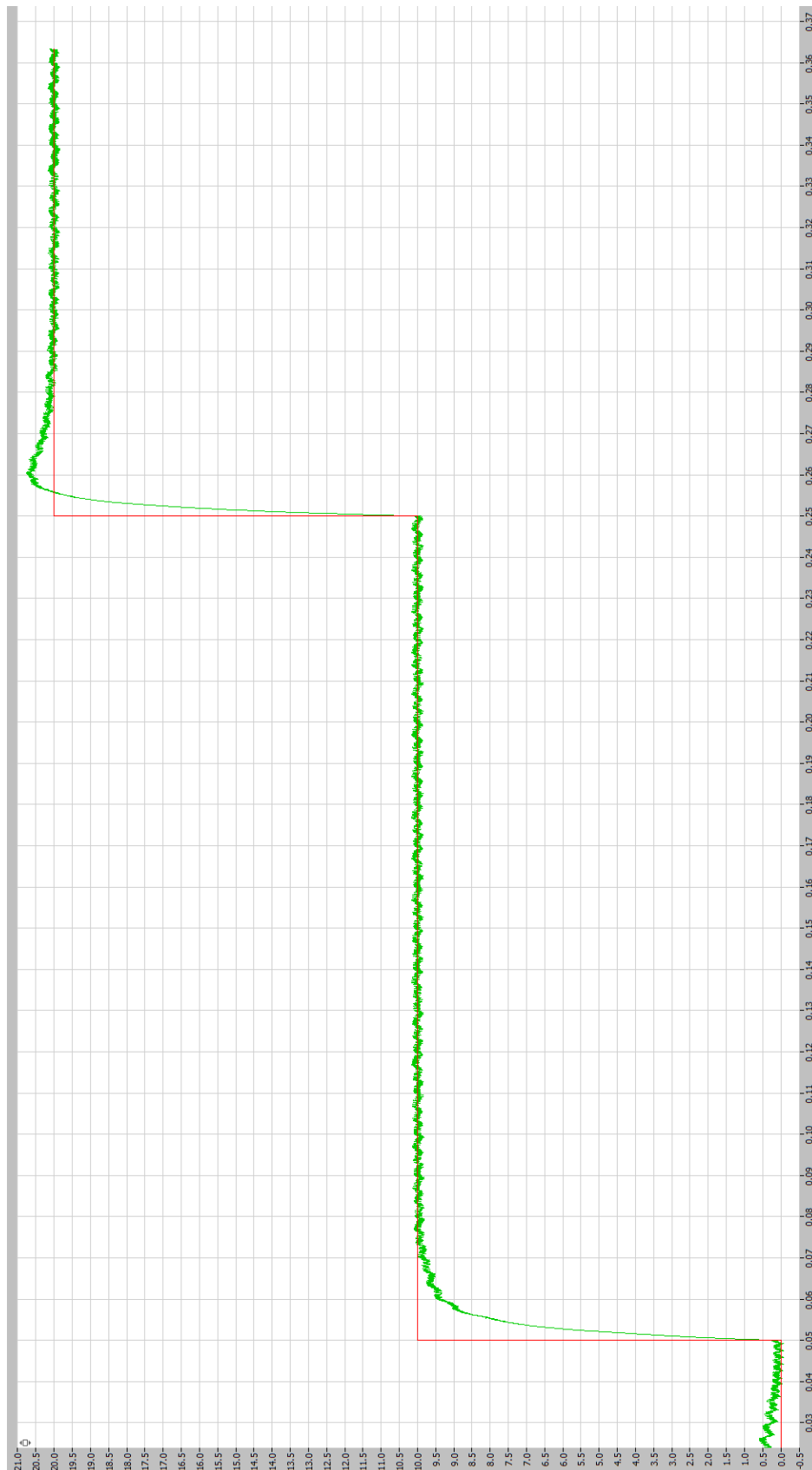


Figure 143 : Figure du courant  $I_{d\_ref}$  avec le courant transmis au réseau  $I_d$

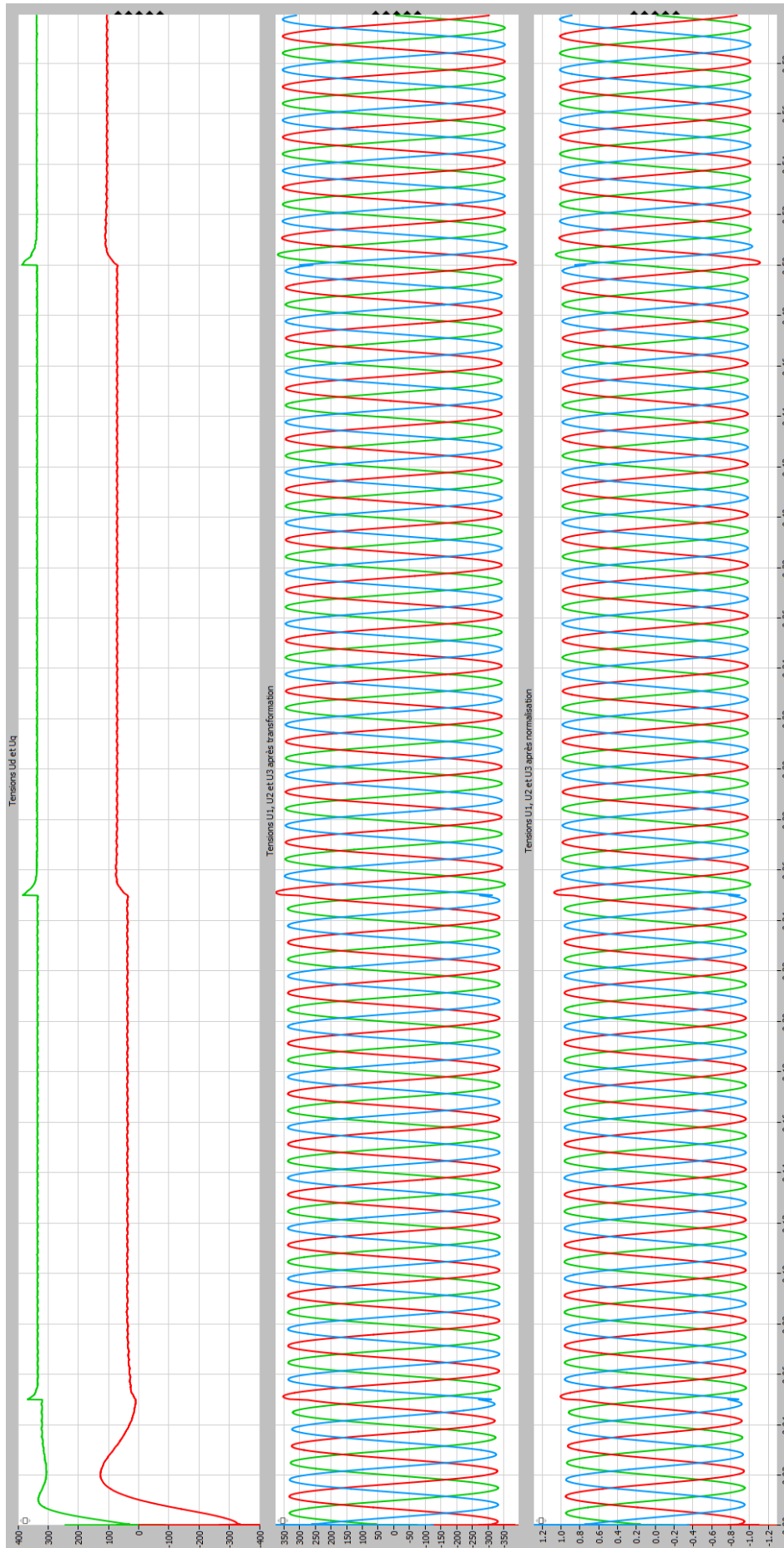


Figure 144 : Transformation de tension, Graphique 1 : Tensions  $U_d$  et  $U_q$ , Graphique 2 : Tension triphasé régulé, Graphique 3 : Tension triphasé normalisé (Figure 46)



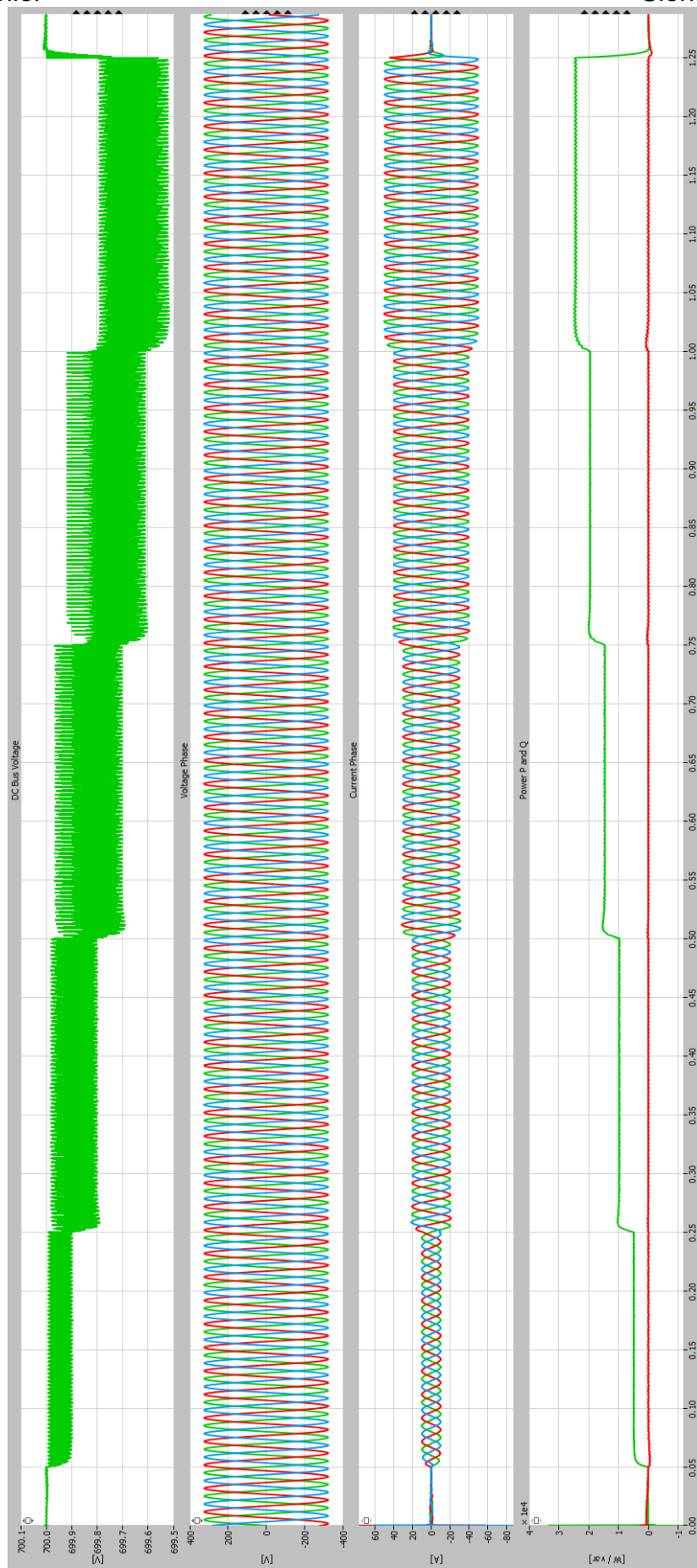


Figure 145 : Graphique 1 : Tension Bus DC, Graphique 2 : Tension de réseau, Graphique 3 : Courants, Graphique 4 : Puissance active et réactive (Figure 47)

### 19.4 Calcul Space Vector Modulation

Secteur →

De 0 à 60° = Secteur 1

De 60 à 120° = Secteur 2

De 120 à 180° = Secteur 3

De 180 à 240° = Secteur 4

De 240 à 300° = Secteur 5

De 300 à 360° = Secteur 6

UDC = Moitié de la tension DC du bus

Vmag = taille du vecteur de référence

Alpha = angle dans chaque secteur. De 0 à 60° et recommence 6 fois par tour

Région →

$$\text{if} \left( v_{mag} \leq \frac{2}{3} * \frac{U_{DC}}{\cos(\alpha) + \frac{1}{\sqrt{3}} * \sin(\alpha)} \right) ==> \text{Region 1}$$

$$\text{else if} \left( v_{mag} * \sin(\alpha) > \frac{2}{3} * \frac{U_{DC} * \sqrt{3}}{2} \right) ==> \text{Region 3}$$

$$\text{else if} \left( v_{mag} * \sin\left(\alpha - \frac{\pi}{3}\right) > \frac{2}{3} * \frac{U_{DC} * \sqrt{3}}{2} \right) ==> \text{Region 4}$$

$$\text{else} ==> \text{Region 2}$$

tau\_a, tau\_b et tau\_c →

Selon la région les calculs sont différents

$$k = \frac{\sqrt{3}}{2} * \frac{v_{mag}}{U_{DC}}$$

Région 1 →

$$\tau_{a} = 2 * k * \sin\left(\frac{\pi}{3} - \alpha\right)$$

$$\tau_{b} = 1 - 2 * k * \sin\left(\frac{\pi}{3} + \alpha\right)$$

$$\tau_{c} = 2 * k * \sin(\alpha)$$

Région 2 →

$$\tau_{a} = 1 - 2 * k * \sin(\alpha)$$

$$\tau_{b} = 2 * k * \sin\left(\frac{\pi}{3} + \alpha\right) - 1$$

$$\tau_{c} = 1 - 2 * k * \sin\left(\frac{\pi}{3} - \alpha\right)$$

Région 3 →

$$\tau_{a} = 2 * k * \sin(\alpha) - 1$$

$$\tau_{b} = 2 * k * \sin\left(\frac{\pi}{3} - \alpha\right)$$

$$\tau_{c} = 2 - 2 * k * \sin\left(\frac{\pi}{3} + \alpha\right)$$

Région 4 →

$$\tau_{a} = 2 - 2 * k * \sin\left(\frac{\pi}{3} + \alpha\right)$$

$$\tau_{b} = 2 * k * \sin(\alpha)$$

$$\tau_{c} = 2 * k * \sin\left(\frac{\pi}{3} - \alpha\right) - 1$$

### 19.5 Vecteurs à choisir pour SVPWM

#### Vecteur à choisir pour le Secteur 1

Multiplie par →	Tau_a	Tau_b	Tau_c
Région 1	V1	V0	V4
Région 2	V1	V2	V4
Région 3	V6	V2	V4
Région 4	V1	V2	V3

#### Vecteur à choisir pour le Secteur 2

Multiplie par →	Tau_a	Tau_b	Tau_c
Région 1	V4	V0	V7
Région 2	V4	V5	V7
Région 3	V9	V5	V7
Région 4	V4	V5	V6

#### Vecteur à choisir pour le Secteur 3

Multiplie par →	Tau_a	Tau_b	Tau_c
Région 1	V7	V0	V10
Région 2	V7	V8	V10
Région 3	V12	V8	V10
Région 4	V7	V8	V9

#### Vecteur à choisir pour le Secteur 4

Multiplie par →	Tau_a	Tau_b	Tau_c
Région 1	V10	V0	V13
Région 2	V10	V11	V13
Région 3	V15	V11	V13
Région 4	V10	V11	V12

#### Vecteur à choisir pour le Secteur 5

Multiplie par →	Tau_a	Tau_b	Tau_c
Région 1	V13	V0	V16
Région 2	V13	V14	V16
Région 3	V18	V14	V16
Région 4	V13	V14	V15

**Vecteur à choisir pour le Secteur 6**

Multiplie par →	Tau_a	Tau_b	Tau_c
Région 1	V16	V0	V1
Région 2	V16	V17	V1
Région 3	V3	V17	V1
Région 4	V16	V17	V18

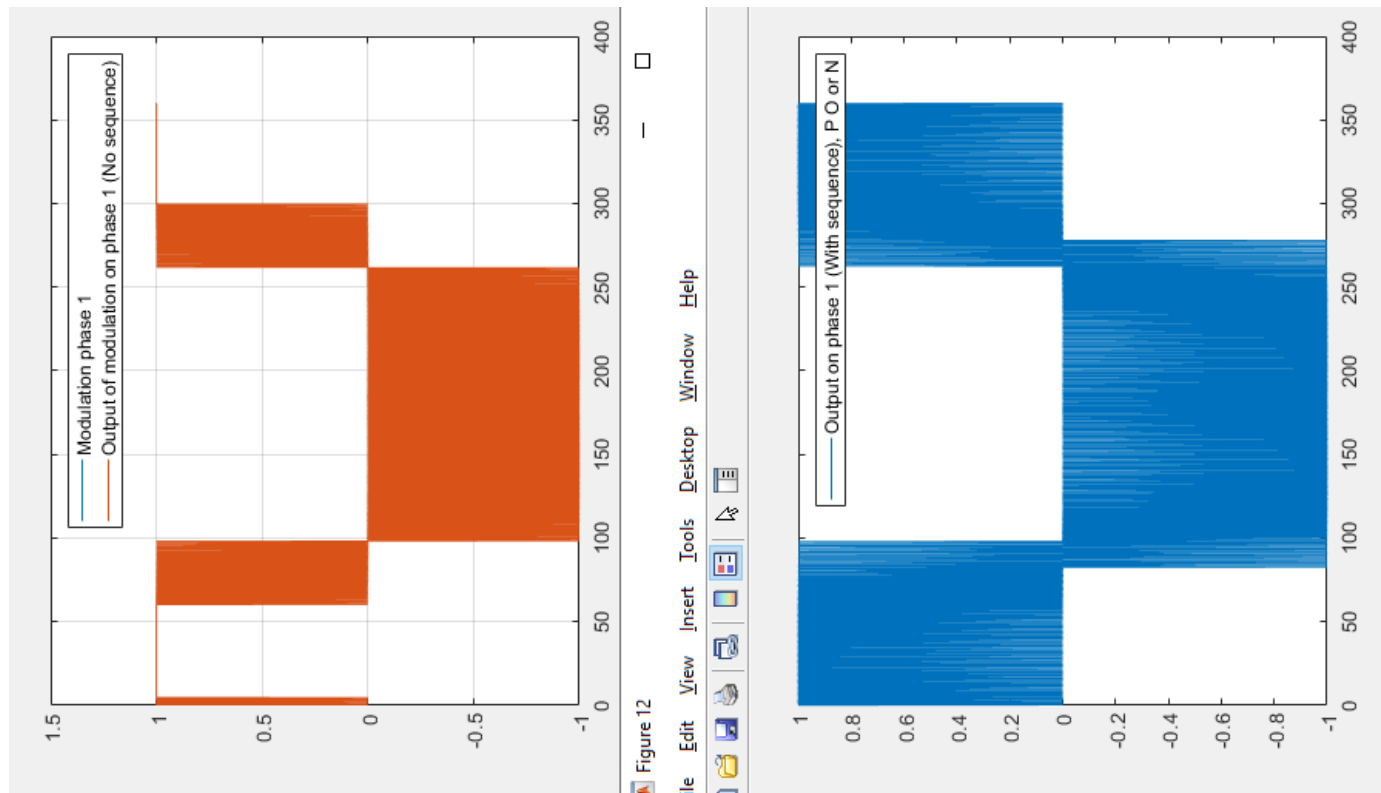
*19.6 Comparaison Matlab entre version SVPWM\_1 et SVPWM\_2*

Figure 146 : Comparaison SVPWM des deux options dans Matlab Phase 1(Figure 38)

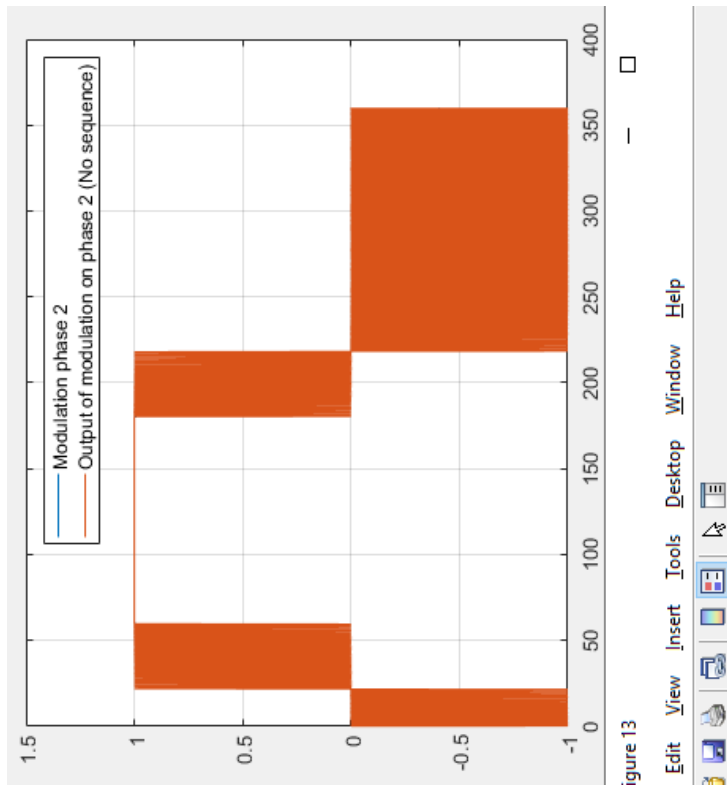


Figure 147 : Comparaison SVPWM des deux options dans Matlab Phase 2 (Figure 38)

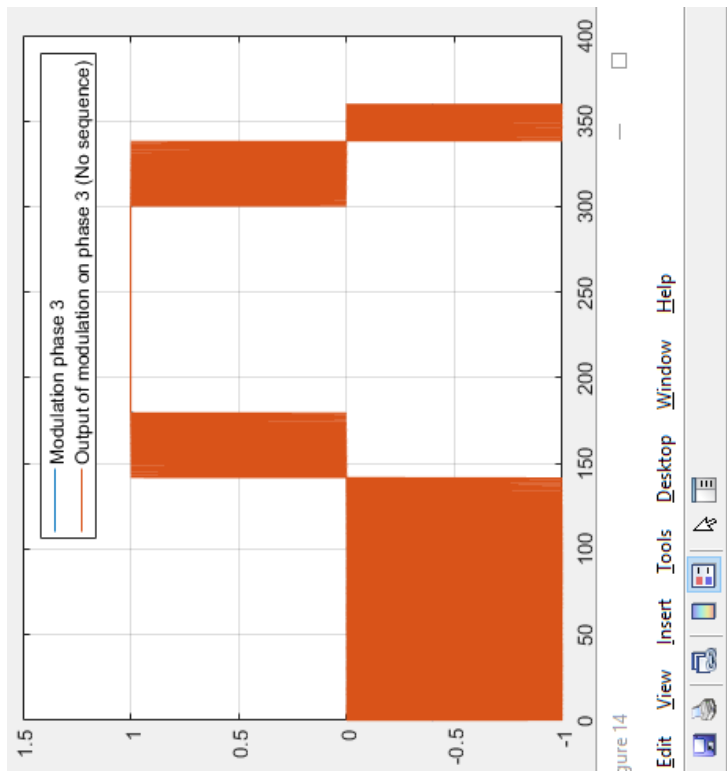
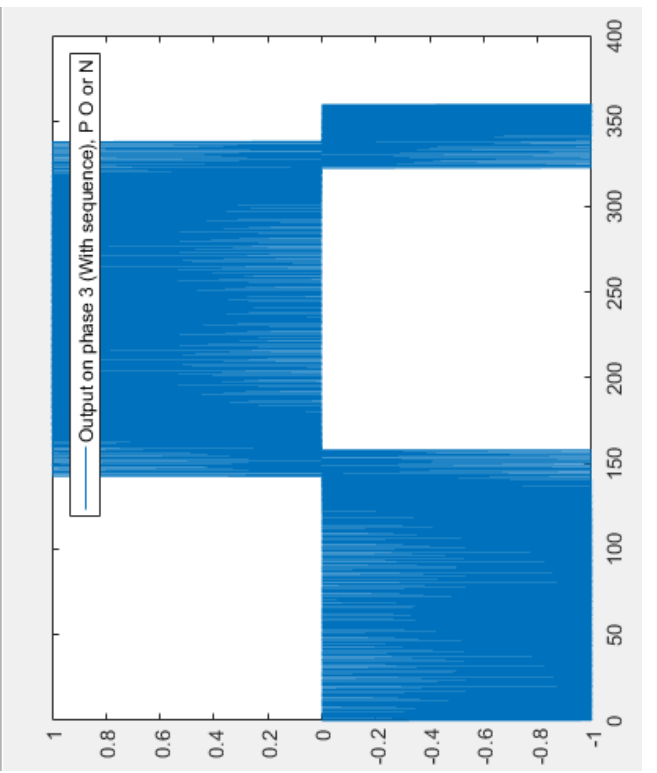
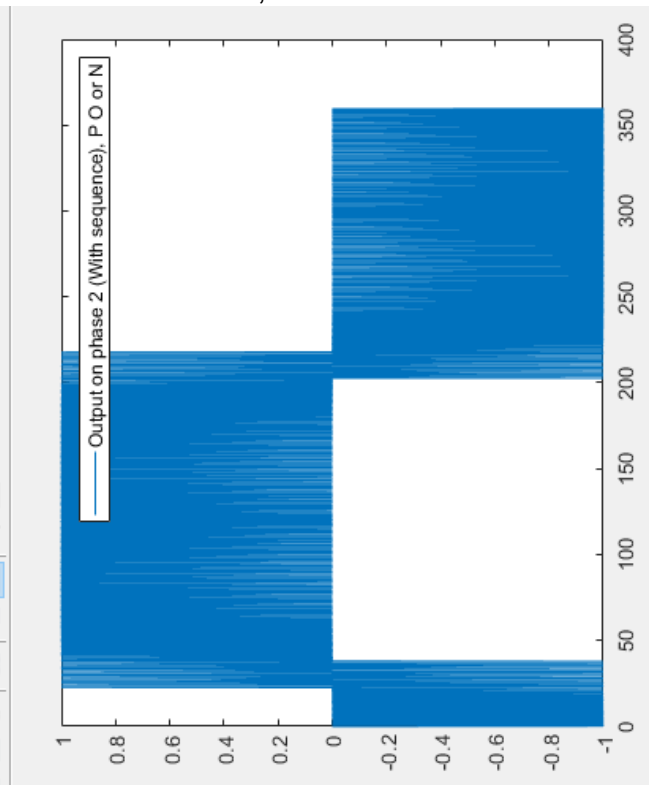


Figure 148 : Comparaison SVPWM des deux options dans Matlab Phase 3 (Figure 38)



### 19.7 Branchement

Ce sont les mêmes branchements pour tous les tests sur la partie Poetic, les différences se font sur les branchements Typhoon ou système réel.



Figure 149 : Rack Poetic avec tous les composants, mais surtout les mesures avec prise Ethernet à gauche et les fibres optiques à droite

Ethernet Poetic :

3 courants AC, 3 tensions AC, 2 tensions DC et la température

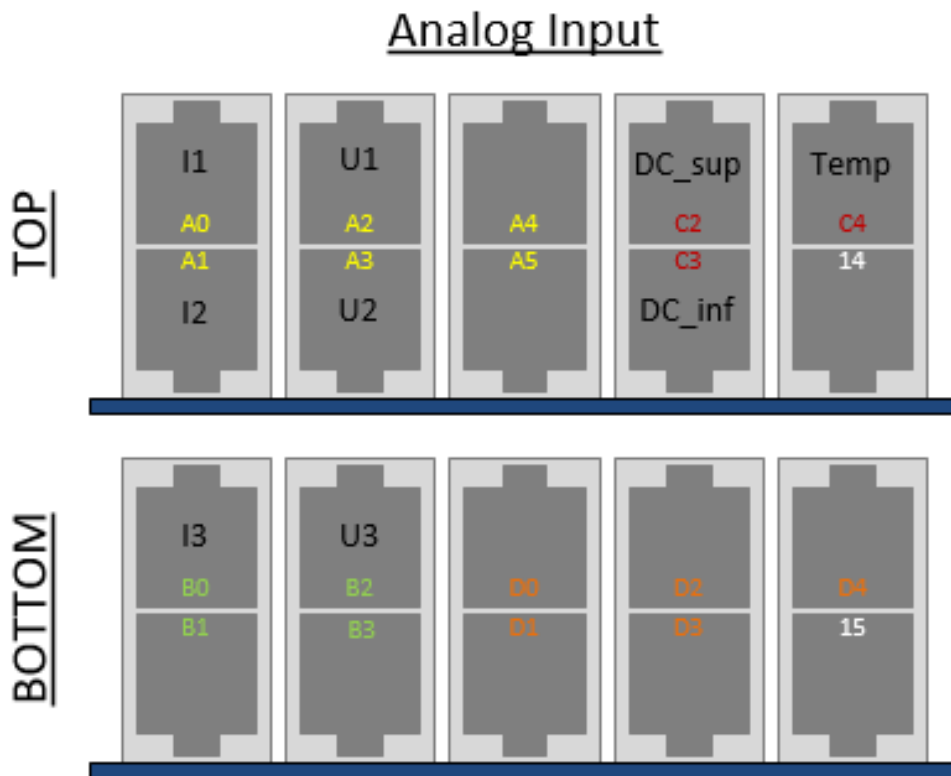


Figure 150 : Analog Input connections Poetic

Fibre optique Poetic :

2 fibres optiques par phases pour un total de 6 fibres optiques + 3 fibres optiques overload. (Il y a des sorties supplémentaires pour activer jusqu'à 5 phases, mais pas utilisé)

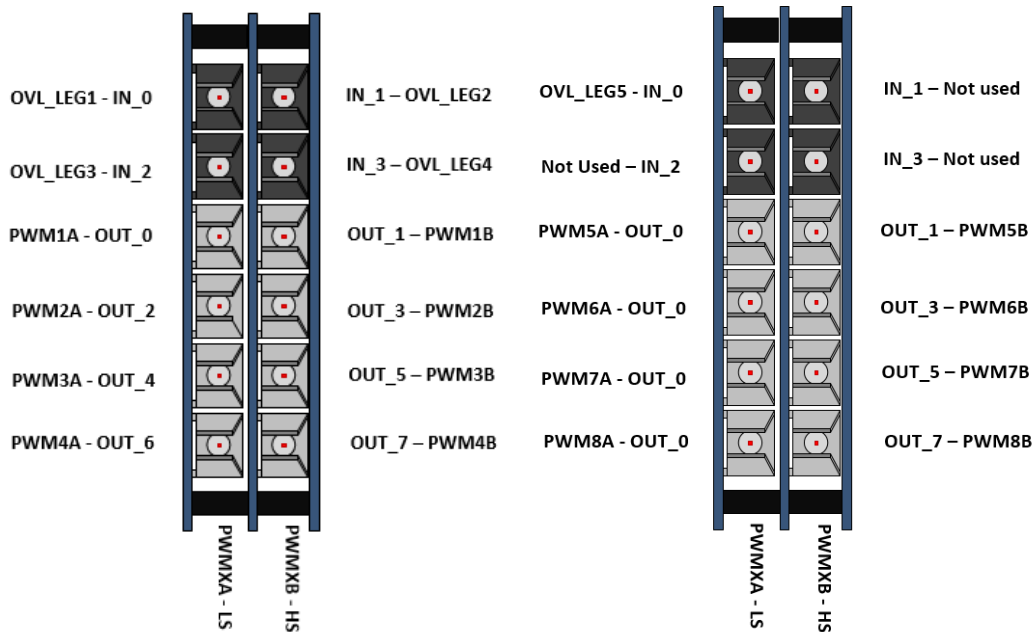


Figure 151 : Cartes fibre optique avec les branchements pour Poetic

### 19.7.1 Poetic + Typhoon

Branchement Typhoon Ethernet :

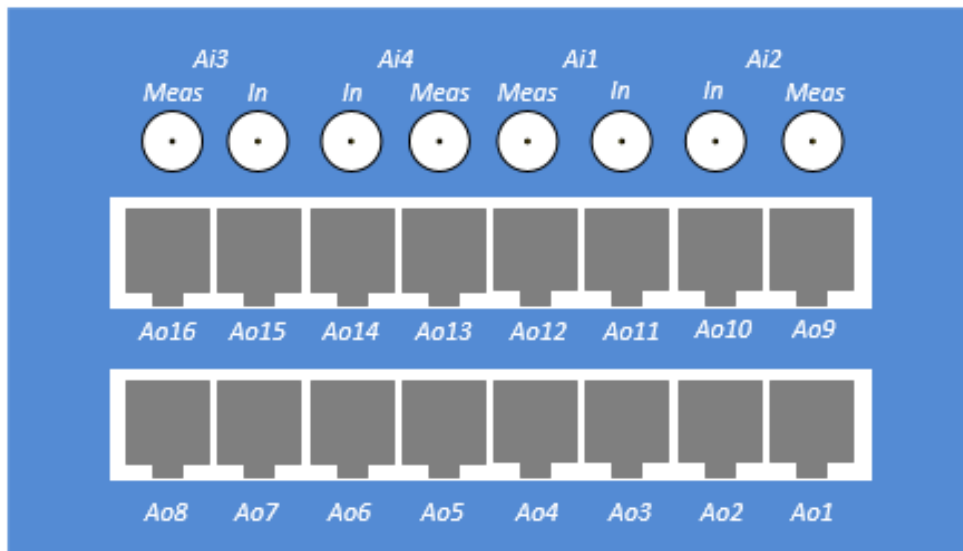


Figure 152 : Entrées et Sorties disponibles sur Typhoon

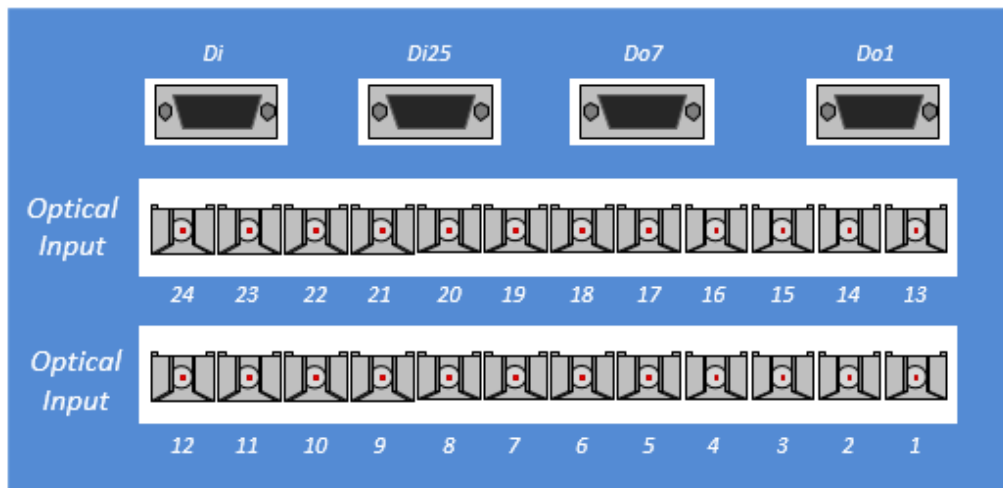


Signal name	Typhoon HIL analog output	POETIC A/D Input
DC Bus Sup	Ao1	C2
I LEG 1	Ao2	A0
I LEG 2	Ao3	A1
I LEG 3	Ao4	B0
U Grid 1	Ao5	A2
U Grid 2	Ao6	A3
U Grid 3	Ao7	B2
DC Bus Inf	Ao8	C3

Figure 153 : Branchement Ethernet pour les mesures sur Typhoon

Branchement Typhoon Fibre optique :

Front



Side



Figure 154 : Les fibres optiques disponibles sur Typhoon

<b>POETIC Optic fiber output</b>	<b>Typhoon HIL digital input</b>
<b>PWM1A</b>	<b>1</b>
<b>PWM1B</b>	<b>2</b>
<b>PWM2A</b>	<b>3</b>
<b>PWM2B</b>	<b>4</b>
<b>PWM3A</b>	<b>5</b>
<b>PWM3B</b>	<b>6</b>
<b>---</b>	<b>7 Not used</b>
<b>---</b>	<b>8 Not used</b>

*Figure 155 : Branchement des Fibres optiques sur les entrées Typhoon*

<b>POETIC Optic fiber input</b>	<b>Typhoon HIL digital output</b>
<b>OVL_LEG1</b>	<b>13</b>
<b>OVL_LEG2</b>	<b>14</b>
<b>OVL_LEG3</b>	<b>15</b>
<b>Unused</b>	<b>16</b>

*Figure 156 : Branchement des Fibres optiques sur les sorties Typhoon*

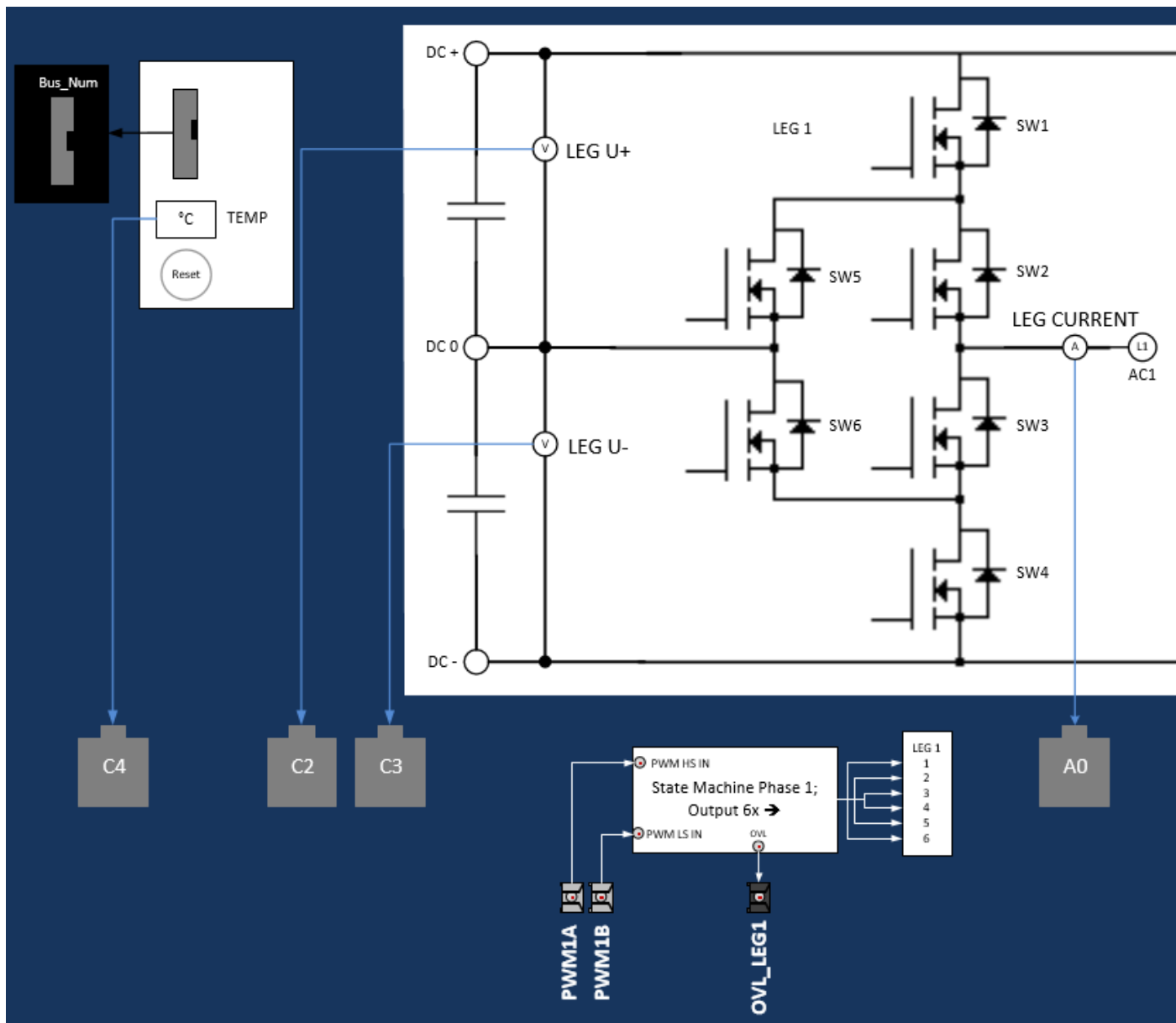


Figure 157 : Panel ANPC Réel (partie 1 gauche)

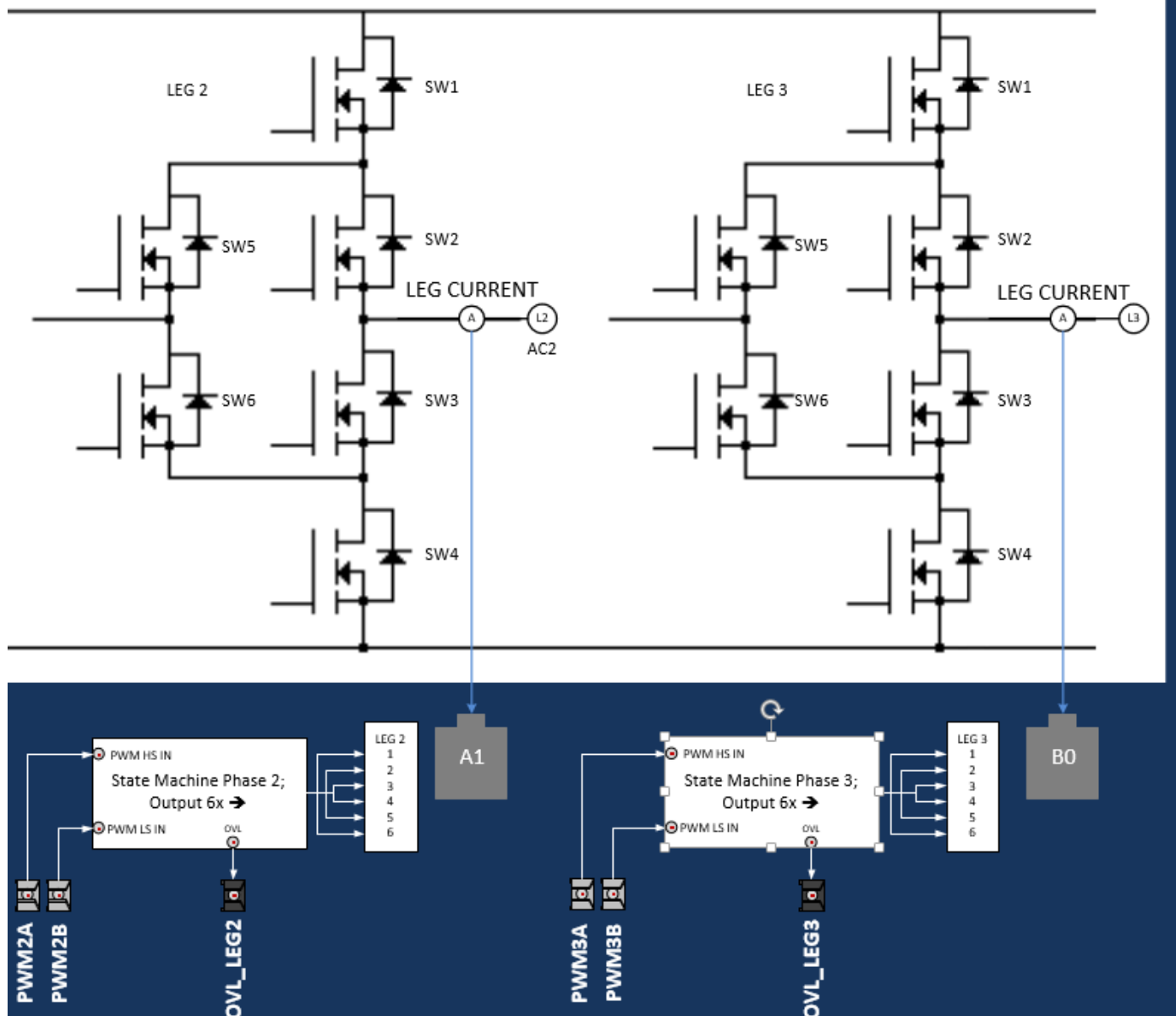


Figure 158 :Panel ANPC réel (partie 2 droite)

Les mesures de tensions ne se font pas sur l'électronique directement. Mais avec un boîtier externe.

#### POETIC Optic fiber output

PWM HS	PWM1A
PWM LS	PWM1B
PWM HS	PWM2A
PWM LS	PWM2B
PWM HS	PWM3A
PWM LS	PWM3B

Figure 159 : Branchement des Fibres optiques avec Typhoon et les entrées réelles

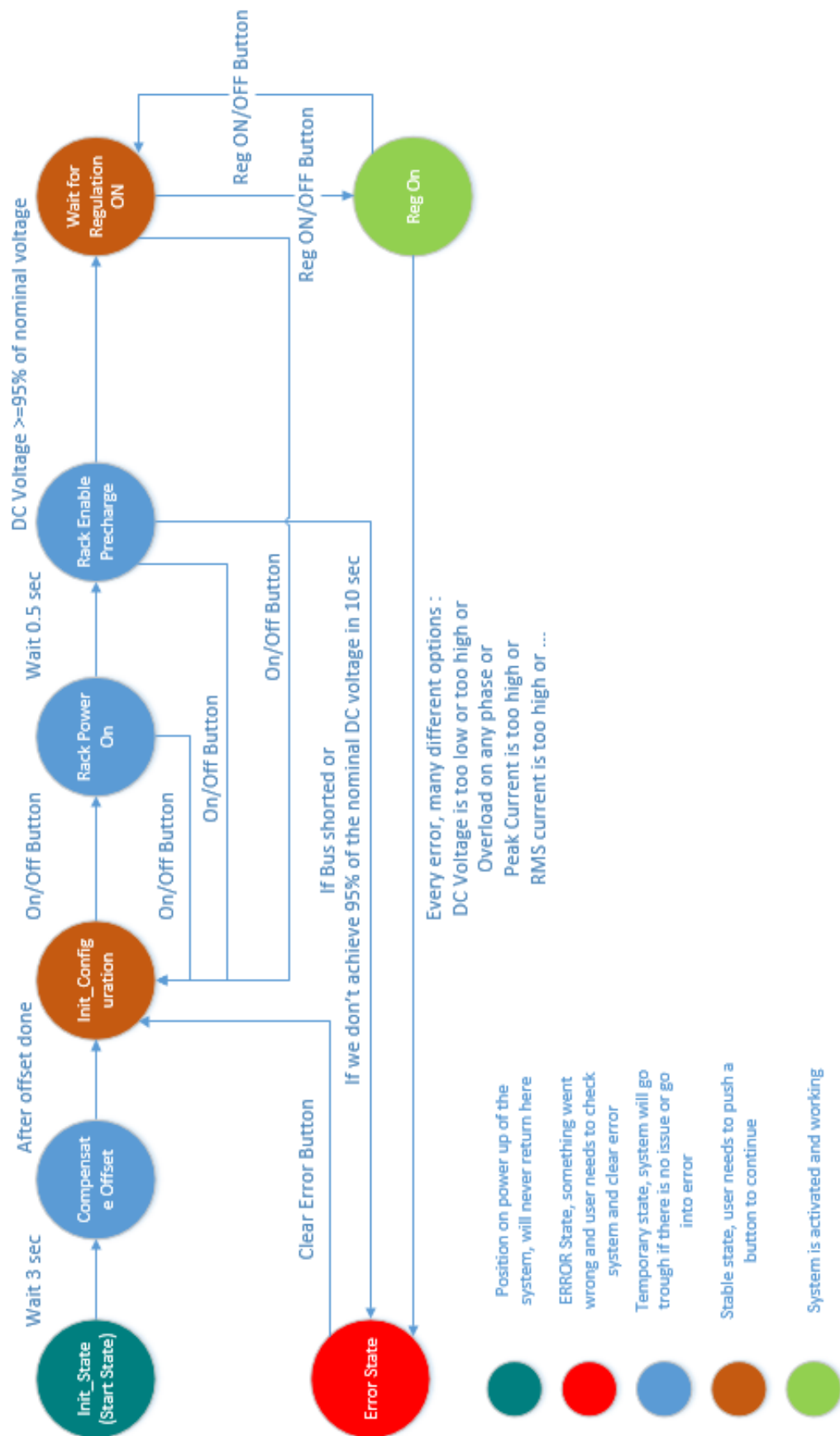


Figure 160 : Machine d'état du système complet (Figure 57)

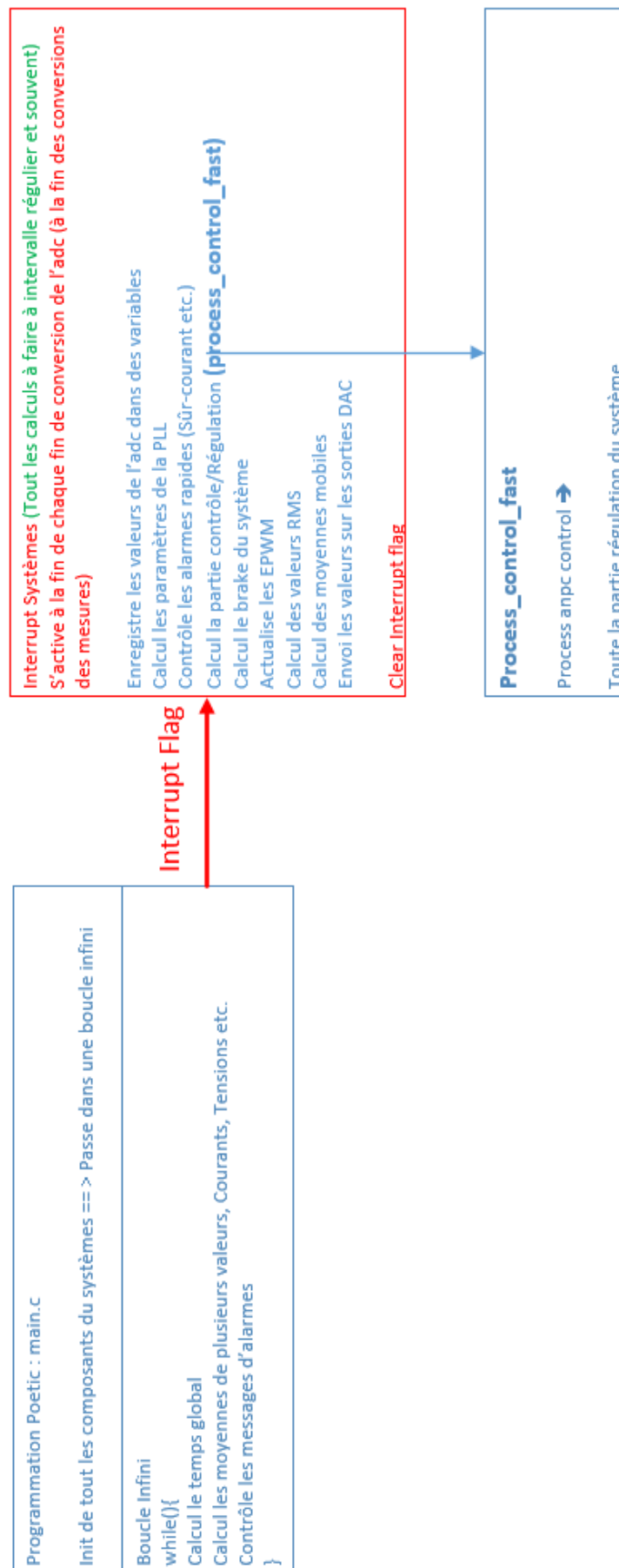


Figure 161 : Fonctionnement général du code Poetic (Figure 58)

## 19.10 Code Space vector modulation

```

41 float vector_ref[19][3][3] =
42 {
43 // {option 1} {option 2} {option 3}
44 //      phase { 1 2 3},{ 1 2 3}
45 { { 0, 0, 0},{ 1, 1, 1},{-1,-1,-1}}, //Vector 0      OOO , PPP , NNN
46 { { 1, 0, 0},{ 0,-1,-1},{ 0, 0, 0}}, //Vector 1      POO , ONN
47 { { 1, 0,-1},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 2      PON
48 { { 1,-1,-1},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 3      PNN
49 { { 1, 1, 0},{ 0, 0,-1},{ 0, 0, 0}}, //Vector 4      PPO , OON
50 { { 0, 1,-1},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 5      ONP
51 { { 1, 1,-1},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 6      PPN
52 { { 0, 1, 0},{-1, 0,-1},{ 0, 0, 0}}, //Vector 7      OPO , NON
53 { {-1, 1, 0},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 8      NPO
54 { {-1, 1,-1},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 9      NPN
55 { { 0, 1, 1},{-1, 0, 0},{ 0, 0, 0}}, //Vector 10     OPP , NOO
56 { {-1, 0, 1},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 11     NOP
57 { {-1, 1, 1},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 12     NPP
58 { { 0, 0, 1},{-1,-1, 0},{ 0, 0, 0}}, //Vector 13     OOP , NNO
59 { { 0,-1, 1},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 14     ONP
60 { {-1,-1, 1},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 15     NNP
61 { { 1, 0, 1},{ 0,-1, 0},{ 0, 0, 0}}, //Vector 16     POP , ONO
62 { { 1,-1, 0},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 17     PNO
63 { { 1,-1, 1},{ 0, 0, 0},{ 0, 0, 0}}, //Vector 18     PNP
64 };

```

Figure 162 : Tous les vecteurs disponibles

```

207 #pragma CODE_SECTION(svpwm_pattern_CPWM_fast, ".TI.ramfunc");
208 void svpwm_three_level(struct dq_to_rst_transf_t *const current_reg_out){
209
210     //We need the value of half of the dc bus voltage for the calculations
211     svthreephase.udc_voltage = (measure.u_bus *0.5);
212     //Save the values of voltage alpha and voltage beta
213     svthreephase.u_alpha = current_reg_out->out_alpha;
214     svthreephase.u_beta  = current_reg_out->out_beta;
215
216     //Calculate the values of size and angle of the reference vector with alpha and beta
217     svthreephase.v_mag = sqrt((svthreephase.u_alpha)*(svthreephase.u_alpha) + (svthreephase.u_beta)*(svthreephase.u_beta));
218     svthreephase.angle_rad = fmod(atan2(svthreephase.u_beta, svthreephase.u_alpha)+2.0*PI,2.0*PI);
219     svthreephase.test = pll.wt - svthreephase.angle_rad;
220     //Find the sector the reference vector is in (Values from 1 to 6)
221     if (svthreephase.angle_rad >= (5.0*PITHIRD)){
222         svthreephase.sector = 6.0;
223     }else if (svthreephase.angle_rad >= (4.0*PITHIRD)){
224         svthreephase.sector = 5.0;
225     }else if (svthreephase.angle_rad >= (3.0*PITHIRD)){
226         svthreephase.sector = 4.0;
227     }else if (svthreephase.angle_rad >= (2.0*PITHIRD)){
228         svthreephase.sector = 3.0;
229     }else if (svthreephase.angle_rad >= (    PITHIRD)){
230         svthreephase.sector = 2.0;
231     }else{
232         svthreephase.sector = 1.0;
233     }
234     //Calculate angle alpha in each sector (from 0 to 59 degrees) and repeats 6 times in every sector
235     //Will be used to calculate region in each sector
236     svthreephase.angle_alpha_rad = (svthreephase.angle_rad - ((svthreephase.sector-1)*(PI/3.0)));
237     svthreephase.angle_alpha_deg = ((svthreephase.angle_alpha_rad/360)*TWOPI);
238 }

```

Figure 163 : Calcul secteur



```

239 //Limit the value of v_mag. Can't be bigger than the Space Vector Hexagon
240 svthreephase.max_mag = ((2.0/3.0) * (svthreephase.udc_voltage*2.0)/(cos(svthreephase.angle_alpha_rad) + 1.0/sqrt(3.0)*sin(svthreephase.angle_alpha_rad)));
241 //We will use the smallest of these 2 values
242 if(svthreephase.max_mag <= svthreephase.v_mag)
243 {
244     svthreephase.v_mag = svthreephase.max_mag;
245     //Bool variable to know if you are at the max value with v_mag
246     svthreephase.max_size_limited = true;
247 } else {svthreephase.max_size_limited = false;}
248
249 //Parameter of the modulation used for calculations (Has nothing to do with the output modulation)
250 svthreephase.modulation = (HALFSQRT3)* (svthreephase.v_mag/svthreephase.udc_voltage) ;
251
252 //Find the region, inside each sector the reference vector is in (Values from 1 to 4)
253 if(svthreephase.v_mag <= (2.0/3.0)*(svthreephase.udc_voltage/(cos(svthreephase.angle_alpha_rad)+(1/sqrt(3.0))*sin(svthreephase.angle_alpha_rad))))
254 {
255     svthreephase.region = 1;
256 }
257 else if ((svthreephase.v_mag * sin(svthreephase.angle_alpha_rad)) > ((2.0/3.0)*svthreephase.udc_voltage*(sqrt(3.0)/2.0)))
258 {
259     svthreephase.region = 3;
260 }
261 else if ((svthreephase.v_mag * sin(PITHIRD-svthreephase.angle_alpha_rad)) > ((2.0/3.0)*svthreephase.udc_voltage*(sqrt(3.0)/2.0)))
262 {
263     svthreephase.region = 4;
264 }
265 else {
266     svthreephase.region = 2;
267 }
268
269 svthreephase.region_f = svthreephase.region;
270 //Save the value of the sector into a uint16_t value. So we have both values if we need them
271 svthreephase.sector_int = (int)svthreephase.sector;

```

Figure 164 : Calcul région

```

273 //Calculate the tau_a, tau_b and tau_c values. The formula is different in each region
274 switch (svthreephase.region){
275     case 1:
276         svthreephase.calculate_tau_a = (2.0 * svthreephase.modulation * sin(PITHIRD - svthreephase.angle_alpha_rad));
277         svthreephase.calculate_tau_b = 1.0 - (2.0 * svthreephase.modulation * sin(PITHIRD + svthreephase.angle_alpha_rad));
278         svthreephase.calculate_tau_c = (2.0 * svthreephase.modulation * sin(svthreephase.angle_alpha_rad));
279         break;
280     case 2:
281         svthreephase.calculate_tau_a = 1.0 - (2.0 * svthreephase.modulation * sin(svthreephase.angle_alpha_rad));
282         svthreephase.calculate_tau_b = (2.0 * svthreephase.modulation * sin(PITHIRD + svthreephase.angle_alpha_rad)) - 1.0;
283         svthreephase.calculate_tau_c = 1.0 - (2.0 * svthreephase.modulation * sin(PITHIRD - svthreephase.angle_alpha_rad));
284         break;
285     case 3:
286         svthreephase.calculate_tau_a = (2.0 * svthreephase.modulation * sin(svthreephase.angle_alpha_rad)) - 1.0;
287         svthreephase.calculate_tau_b = (2.0 * svthreephase.modulation * sin(PITHIRD - svthreephase.angle_alpha_rad));
288         svthreephase.calculate_tau_c = 2.0 - (2.0 * svthreephase.modulation * sin(PITHIRD + svthreephase.angle_alpha_rad));
289         break;
290     case 4:
291         svthreephase.calculate_tau_a = 2.0 - (2.0 * svthreephase.modulation * sin(PITHIRD + svthreephase.angle_alpha_rad));
292         svthreephase.calculate_tau_b = (2.0 * svthreephase.modulation * sin(svthreephase.angle_alpha_rad));
293         svthreephase.calculate_tau_c = (2.0 * svthreephase.modulation * sin(PITHIRD - svthreephase.angle_alpha_rad)) - 1.0;
294         break;
295     default:
296         break;
297 }
298
299 //Save the calculated tau values into these variables
300 svthreephase.tau_a = svthreephase.calculate_tau_a;
301 svthreephase.tau_b = svthreephase.calculate_tau_b;
302 svthreephase.tau_c = svthreephase.calculate_tau_c;
303

```

Figure 165 : Calcul tau\_a, tau\_b et tau\_c

```

304 //Calculate a parameter we will use
305 //This parameter makes us chose the right vector in each sector. Changing the sector means we need to rotate every vector by 60 deg. The new vectors in the next sector will always be 3 position after the first we used.
306 //Example : In sector 1 we will chose vector 1 and in the next sector the "same vector" will be vector 4 and the next will be 7 and so on.
307 //This formula will calculate the right vector we need
308 svthreephase.vector_param_1 = 1+3*(svthreephase.sector_int-1);
309 svthreephase.vector_param_2 = 2+3*(svthreephase.sector_int-1);
310 svthreephase.vector_param_3 = 3+3*(svthreephase.sector_int-1);
311 svthreephase.vector_param_4 = 4+3*(svthreephase.sector_int-1);
312 svthreephase.vector_param_6 = 6+3*(svthreephase.sector_int-1);
313
314 //In the last sector we will need to use the first vectors again.
315 //So for the parameters 4 and 6 we will need to use the vectors 1 and vector 3 again
316 //and not the vectors 19 and 21 like the formula above calculates.
317
318 //Limits the value of vectors we use
319 if(svthreephase.vector_param_4>18){
320     svthreephase.vector_param_4 -= 18;
321 }
322 if(svthreephase.vector_param_6>18){
323     svthreephase.vector_param_6 -= 18;
324 }
325
326 //Balance DC Bus
327 //If there are uneven voltages in the upper and lower capacitor we swap to the other option with the reference vectors.
328 svthreephase.balance = measure.u_bus_sup-measure.u_bus_inf;
329 //If DC Bus is using more the lower capacitor. Then the upper capacitor will have a bigger voltage and we will swap to use the P and O Vectors more.
330 if(svthreephase.balance>50.0){
331     svthreephase.dc_bus_balancing = 0;
332 }
333 //If DC Bus is using more the upper capacitor. Then the lower capacitor will have a bigger voltage and we will swap to use the N and O Vectors more.
334 if(-svthreephase.balance>50.0){
335     svthreephase.dc_bus_balancing = 1;
336 }
337 //If we are not in any of these 2 cases we don't change the value of dc_bus_balancing. "Schmitt Trigger"

```

Figure 166 : Quels vecteurs choisir

```

340 //Calculate the modulation with the reference vectors and the tau_a, tau_b and tau_c values
341 //svthreephase.dc_bus_balancing" is used to change to other choice of the vector, some vectors have 2 options for the same result. We use this variable to balance the DC_Bus.
342 //WARNING ONLY VECTOR PARAM 1 AND 4 HAVE OTHER VECTORS AS OPTION. VECTOR PARAM 2 WILL ALWAYS BE IN OPTION 0
343 switch (svthreephase.region) {
344     case 1:
345         //vector_ref = all vectors available[ vector by sector & region ][ alternative vector choice ][phase]
346         svthreephase.phase_1_v1 = vector_ref[svthreephase.vector_param_1][svthreephase.dc_bus_balancing][0] * svthreephase.tau_a;
347         svthreephase.phase_2_v1 = vector_ref[svthreephase.vector_param_1][svthreephase.dc_bus_balancing][1] * svthreephase.tau_a;
348         svthreephase.phase_3_v1 = vector_ref[svthreephase.vector_param_1][svthreephase.dc_bus_balancing][2] * svthreephase.tau_a;
349         svthreephase.phase_1_v2 = vector_ref[0][0] * svthreephase.tau_b;
350         svthreephase.phase_2_v2 = vector_ref[0][1] * svthreephase.tau_b;
351         svthreephase.phase_3_v2 = vector_ref[0][2] * svthreephase.tau_b;
352         svthreephase.phase_1_v3 = vector_ref[svthreephase.vector_param_4][svthreephase.dc_bus_balancing][0] * svthreephase.tau_c;
353         svthreephase.phase_2_v3 = vector_ref[svthreephase.vector_param_4][svthreephase.dc_bus_balancing][1] * svthreephase.tau_c;
354         svthreephase.phase_3_v3 = vector_ref[svthreephase.vector_param_4][svthreephase.dc_bus_balancing][2] * svthreephase.tau_c;
355         break;
356     case 2:
357         //vector_ref = all vectors available[ vector by region, sector ][ alternative vector choice ][phase]
358         svthreephase.phase_1_v1 = vector_ref[svthreephase.vector_param_1][svthreephase.dc_bus_balancing][0] * svthreephase.tau_a;
359         svthreephase.phase_2_v1 = vector_ref[svthreephase.vector_param_1][svthreephase.dc_bus_balancing][1] * svthreephase.tau_a;
360         svthreephase.phase_3_v1 = vector_ref[svthreephase.vector_param_1][svthreephase.dc_bus_balancing][2] * svthreephase.tau_a;
361         svthreephase.phase_1_v2 = vector_ref[svthreephase.vector_param_2][0][0] * svthreephase.tau_b;
362         svthreephase.phase_2_v2 = vector_ref[svthreephase.vector_param_2][0][1] * svthreephase.tau_b;
363         svthreephase.phase_3_v2 = vector_ref[svthreephase.vector_param_2][0][2] * svthreephase.tau_b;
364         svthreephase.phase_1_v3 = vector_ref[svthreephase.vector_param_4][svthreephase.dc_bus_balancing][0] * svthreephase.tau_c;
365         svthreephase.phase_2_v3 = vector_ref[svthreephase.vector_param_4][svthreephase.dc_bus_balancing][1] * svthreephase.tau_c;
366         svthreephase.phase_3_v3 = vector_ref[svthreephase.vector_param_4][svthreephase.dc_bus_balancing][2] * svthreephase.tau_c;
367         break;
368

```

Figure 167 : Multiplier les vecteurs avec les valeurs de tau (partie 1)

```

370
371 //vector_ref = all vectors available[ vector by region, sector ][[ alternative vector choice ][[phase]
372 svthreephase.phase_1_v1 = vector_ref[svthreephase.vector_param_6][0][0] * svthreephase.tau_a;
373 svthreephase.phase_2_v1 = vector_ref[svthreephase.vector_param_6][0][1] * svthreephase.tau_a;
374 svthreephase.phase_3_v1 = vector_ref[svthreephase.vector_param_6][0][2] * svthreephase.tau_a;
375 svthreephase.phase_1_v2 = vector_ref[svthreephase.vector_param_2][0][0] * svthreephase.tau_b;
376 svthreephase.phase_2_v2 = vector_ref[svthreephase.vector_param_2][0][1] * svthreephase.tau_b;
377 svthreephase.phase_3_v2 = vector_ref[svthreephase.vector_param_2][0][2] * svthreephase.tau_b;
378 svthreephase.phase_1_v3 = vector_ref[svthreephase.vector_param_4][svthreephase.dc_bus_balancing][0] * svthreephase.tau_c;
379 svthreephase.phase_2_v3 = vector_ref[svthreephase.vector_param_4][svthreephase.dc_bus_balancing][1] * svthreephase.tau_c;
380 svthreephase.phase_3_v3 = vector_ref[svthreephase.vector_param_4][svthreephase.dc_bus_balancing][2] * svthreephase.tau_c;
381 break;
382
383
384 case 4:
385 //vector_ref = all vectors available[ vector by region, sector ][[ alternative vector choice ][[phase]
386 svthreephase.phase_1_v1 = vector_ref[svthreephase.vector_param_1][svthreephase.dc_bus_balancing][0] * svthreephase.tau_a;
387 svthreephase.phase_2_v1 = vector_ref[svthreephase.vector_param_1][svthreephase.dc_bus_balancing][1] * svthreephase.tau_a;
388 svthreephase.phase_3_v1 = vector_ref[svthreephase.vector_param_1][svthreephase.dc_bus_balancing][2] * svthreephase.tau_a;
389 svthreephase.phase_1_v2 = vector_ref[svthreephase.vector_param_2][0][0] * svthreephase.tau_b;
390 svthreephase.phase_2_v2 = vector_ref[svthreephase.vector_param_2][0][1] * svthreephase.tau_b;
391 svthreephase.phase_3_v2 = vector_ref[svthreephase.vector_param_2][0][2] * svthreephase.tau_b;
392 svthreephase.phase_1_v3 = vector_ref[svthreephase.vector_param_3][0][0] * svthreephase.tau_c;
393 svthreephase.phase_2_v3 = vector_ref[svthreephase.vector_param_3][0][1] * svthreephase.tau_c;
394 svthreephase.phase_3_v3 = vector_ref[svthreephase.vector_param_3][0][2] * svthreephase.tau_c;
395 break;
396
397 default:
398 break;
399 }

```

Figure 168 : Multiplier les vecteurs avec les valeurs de tau (partie 2)

```

400 //Output in case anybody needs this version, value between (-1 and 1). But for our system ANPC, 2 fiber optics transmit data so we need a different output
401 svthreephase.out_modulation_phase_1 = svthreephase.phase_1_v1 + svthreephase.phase_1_v2 + svthreephase.phase_1_v3;
402 svthreephase.out_modulation_phase_2 = svthreephase.phase_2_v1 + svthreephase.phase_2_v2 + svthreephase.phase_2_v3;
403 svthreephase.out_modulation_phase_3 = svthreephase.phase_3_v1 + svthreephase.phase_3_v2 + svthreephase.phase_3_v3;
404
405 //Save in another variable to continue the transformation of the output
406 svthreephase.modulation_phase_1 = svthreephase.out_modulation_phase_1;
407 svthreephase.modulation_phase_2 = svthreephase.out_modulation_phase_2;
408 svthreephase.modulation_phase_3 = svthreephase.out_modulation_phase_3;
409
410 //We limit the signal even if the modulator already does this
411 svthreephase.modulation_phase_1_A = ( svthreephase.modulation_phase_1);
412 LIMIT(svthreephase.modulation_phase_1_A,0.0,1.0);
413 svthreephase.modulation_phase_1_B = (-svthreephase.modulation_phase_1);
414 LIMIT(svthreephase.modulation_phase_1_B,0.0,1.0);
415 svthreephase.modulation_phase_2_A = ( svthreephase.modulation_phase_2);
416 LIMIT(svthreephase.modulation_phase_2_A,0.0,1.0);
417 svthreephase.modulation_phase_2_B = (-svthreephase.modulation_phase_2);
418 LIMIT(svthreephase.modulation_phase_2_B,0.0,1.0);
419 svthreephase.modulation_phase_3_A = ( svthreephase.modulation_phase_3);
420 LIMIT(svthreephase.modulation_phase_3_A,0.0,1.0);
421 svthreephase.modulation_phase_3_B = (-svthreephase.modulation_phase_3);
422 LIMIT(svthreephase.modulation_phase_3_B,0.0,1.0);

```

*Figure 169 : Addition des trois vecteurs pour reproduire le vecteur référence avec les trois phases*

```

424 //If the modulation is 0. We set the modulation state to 0.0
425 //If the modulation is negative we set the modulation state to -1.0 and invert the signal.
426 //And else we set the modulation state to 1.0
427 //
428 //Modulation on phase 1
429 if(svthreephase.modulation_phase_1==0)
430 {
431     svthreephase.phase_1_state = 0.0;
432     svthreephase.out_modulation_phase_1_on_A = 1.0;
433     svthreephase.out_modulation_phase_1_on_B = 1.0;
434 }
435 else if(svthreephase.modulation_phase_1<0)
436 {
437     svthreephase.phase_1_state = -1.0;
438     svthreephase.out_modulation_phase_1_on_A = 1.0;
439     svthreephase.out_modulation_phase_1_on_B = 1.0 - svthreephase.modulation_phase_1_B;
440 }
441 else
442 {
443     svthreephase.phase_1_state = 1.0;
444     svthreephase.out_modulation_phase_1_on_A = 1.0 - svthreephase.modulation_phase_1_A;
445     svthreephase.out_modulation_phase_1_on_B = 1.0;
446 }

448 //Modulation on phase 2
449 if(svthreephase.modulation_phase_2==0)
450 {
451     svthreephase.phase_2_state = 0.0;
452     svthreephase.out_modulation_phase_2_on_A = 1.0;
453     svthreephase.out_modulation_phase_2_on_B = 1.0;
454 }
455 else if(svthreephase.modulation_phase_2<0)
456 {
457     svthreephase.phase_2_state = -1.0;
458     svthreephase.out_modulation_phase_2_on_A = 1.0;
459     svthreephase.out_modulation_phase_2_on_B = 1.0 - svthreephase.modulation_phase_2_B;
460 }
461 else
462 {
463     svthreephase.phase_2_state = 1.0;
464     svthreephase.out_modulation_phase_2_on_A = 1.0 - svthreephase.modulation_phase_2_A;
465     svthreephase.out_modulation_phase_2_on_B = 1.0;
466 }
467
468 //Modulation on phase 3
469 if(svthreephase.modulation_phase_3==0)
470 {
471     svthreephase.phase_3_state = 0.0;
472     svthreephase.out_modulation_phase_3_on_A = 1.0;
473     svthreephase.out_modulation_phase_3_on_B = 1.0;
474 }
475 else if(svthreephase.modulation_phase_3<0)
476 {
477     svthreephase.phase_3_state = -1.0;
478     svthreephase.out_modulation_phase_3_on_A = 1.0;
479     svthreephase.out_modulation_phase_3_on_B = 1.0 - svthreephase.modulation_phase_3_B;
480 }
481 else
482 {
483     svthreephase.phase_3_state = 1.0;
484     svthreephase.out_modulation_phase_3_on_A = 1.0 - svthreephase.modulation_phase_3_A;
485     svthreephase.out_modulation_phase_3_on_B = 1.0;
486 }
487 }
488 }
489 //End Space Vector Modulation
---
```

Figure 170 : Valeurs à envoyer aux modulateurs



## 19.11 Résultat Space Vector Modulation spécial

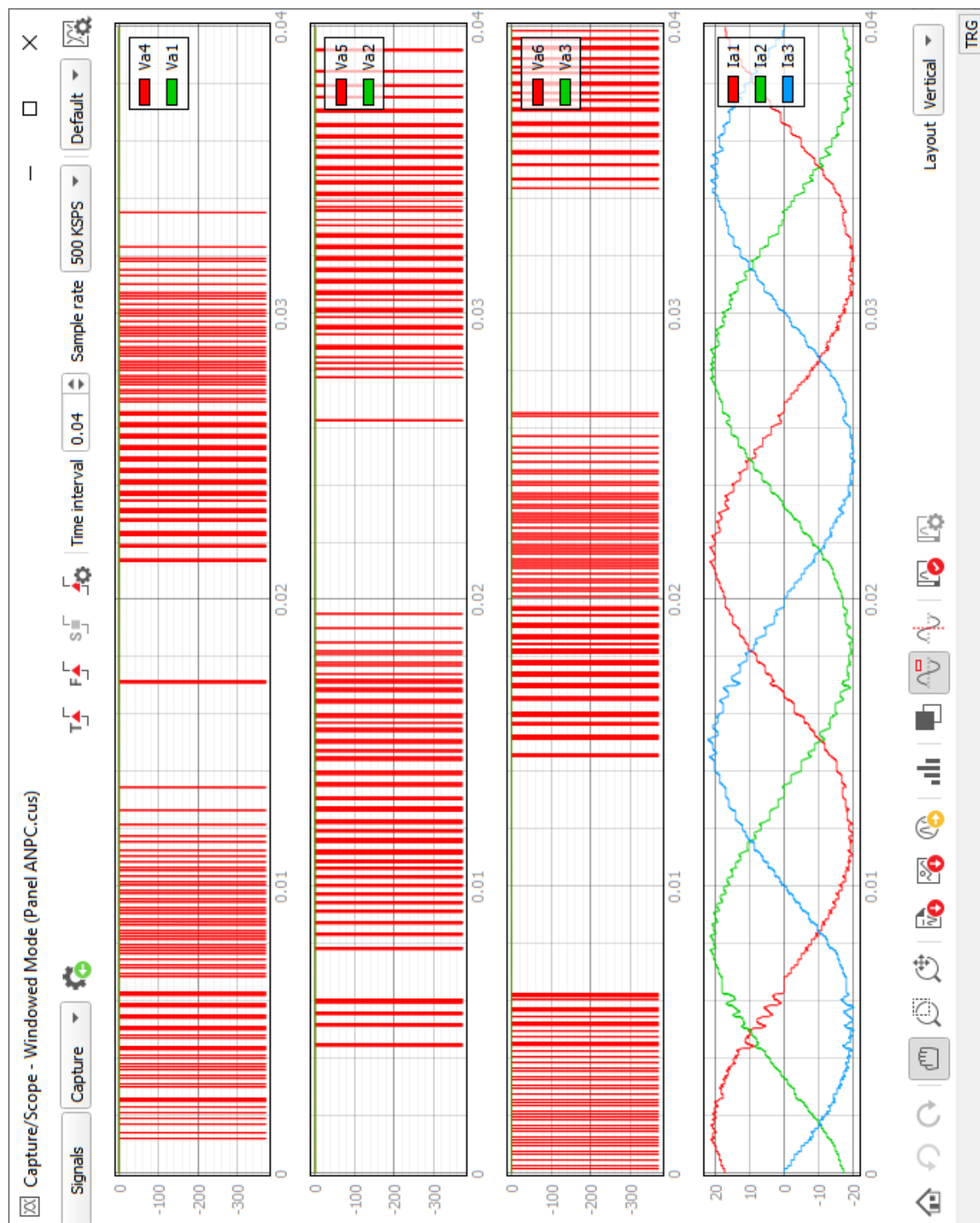


Figure 171 : Fonctionnement en Space Vector Modulation avec uniquement des tensions négatives (Figure 87)

19.12 Transmission des données

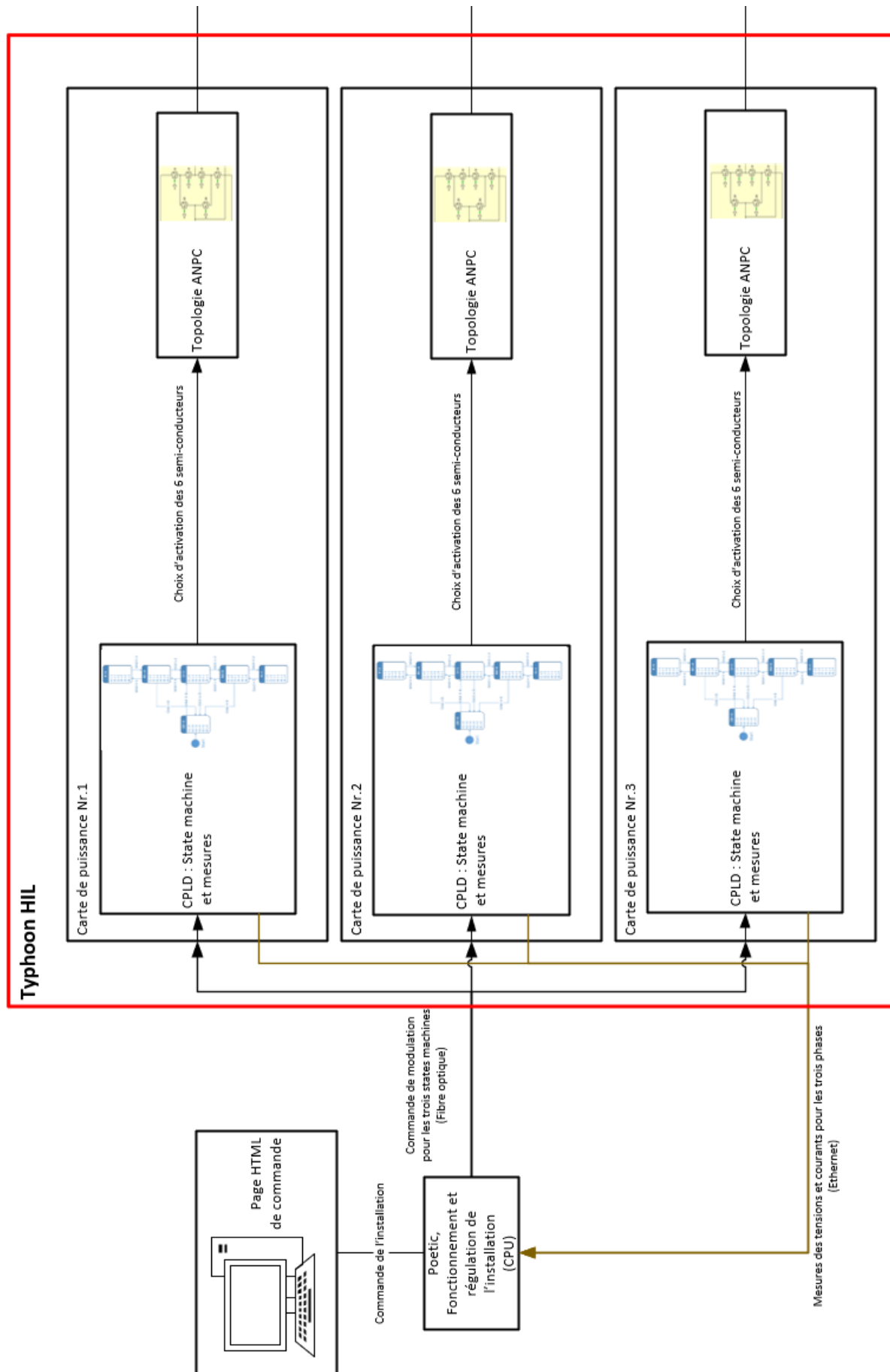


Figure 172 : Images du système avec plus de détails (Figure 27)

## 19.13 Mode d'emploi pour activer l'installation

### 19.13.1 Poetic + Typhoon

Pour faire les tests avec Typhoon il faut commencer par avoir le bon câblage sur l'installation. Les branchements sont disponibles au chapitre 19.7 en annexe

1. Alimentation du rack avec la multiprise
2. Allume Typhoon et la partie Poetic avec les switch's à l'arrière
3. Sur le PC, se connecter à Typhoon avec la bonne adresse IP
4. Charge le bon programme de simulation
5. Une fois que le Panel s'ouvre, il faut charger les paramètres d'entrées et sorties → Figure 173 (les paramètres dans Analog et Digital Output sont disponible à la Figure 68 si le fichier n'est pas disponible)

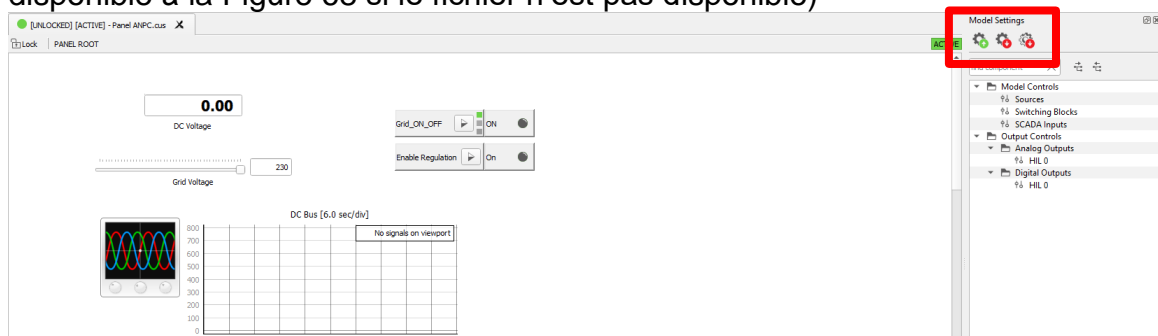


Figure 173 : Charger les entrées et sorties du système, avec le bouton vert

6. Sur le PC avec code composer, il faut 'build' le bon programme pour le CPU
7. Ouvrir le programme 'dspctrl\_prog\_tools.dvx', qui permet de charger dans le CPU le programme



Figure 174 : Charger le programme dans le CPU

8. Ouvrir le programme 'CmdANPC\_Interface\_base.dvx' qui est l'interface de commande du système → Figure 64
9. Il faut lancer la simulation Typhoon (A la fin car l'offset se calcule au lancement du programme Poetic.
10. Le système est maintenant prêt, il faut utiliser l'interface pour le pilotage
11. (Aide pour le pilotage du système à la Figure 57)

### 19.13.2 Poetic + Système réel

Pour faire les tests avec Poetic et le système réel il faut commencer par avoir le bon câblage sur l'installation. Les branchements sont disponibles au chapitre 19.7 en annexe.

1. Alimentation du rack avec la multiprise
2. Brancher l'alimentation DC 1500V avec la prise triphasée
3. Allume l'alimentation DC avec le switch à l'avant (Mettre sur 0V et OFF)
4. Allume la partie Poetic et électronique avec les switch's à l'arrière
5. Sur le PC avec code composer, il faut 'build' le bon programme pour le CPU
6. Ouvrir le programme 'dspctrl\_prog\_tools.dvx', qui permet de charger dans le CPU le programme, brancher le câble USB à l'avant de POETIC



Figure 175 : Charger le programme dans le CPU

7. Ouvrir le programme 'CmdANPC\_Interface\_base.dvx' qui est l'interface de commande du système → Figure 64
8. Mettre l'alimentation sur ON et augmenter la tension pour atteindre la tension voulu pour le test
9. Le système est prêt, il faut utiliser l'interface pour le pilotage