

CHAPITRE 2

Algorithme et plateforme

[MCours.com](https://www.mycours.com)

2.1 Introduction et schématisation de l'algorithme

Avant d'amorcer le développement du prototype, il faut étudier l'algorithme utilisé. Dans le cas présent, il s'agit de la commande vectorielle. La recherche bibliographique permet d'identifier des articles et des sources qui expliquent la commande afin d'en tirer les équations mathématiques. On peut alors schématiser la commande vectorielle sous forme de boîtes connectées par des fils. Chacun de ces blocs va produire des sorties en appliquant les entrées aux équations du bloc.

2.1.1 Contrôle vectoriel

Le choix du contrôle vectoriel est intéressant comme application utilisée afin de créer une méthode de prototypage optimale. C'est sa complexité et sa modularité qui offrent un défi intéressant ainsi qu'un bon terrain d'évaluation pour des outils de co-simulation. La présence d'une racine carrée et d'une division offre aussi l'avantage de pouvoir appliquer et vérifier l'implémentation de ces opérations selon un modèle non restaurant plus économique en terme de ressources logiques (voir section 3.4). Les résultats de l'implémentation de ces opérations ont permis de publier deux articles sur le sujet que l'on retrouve à l'annexe I.

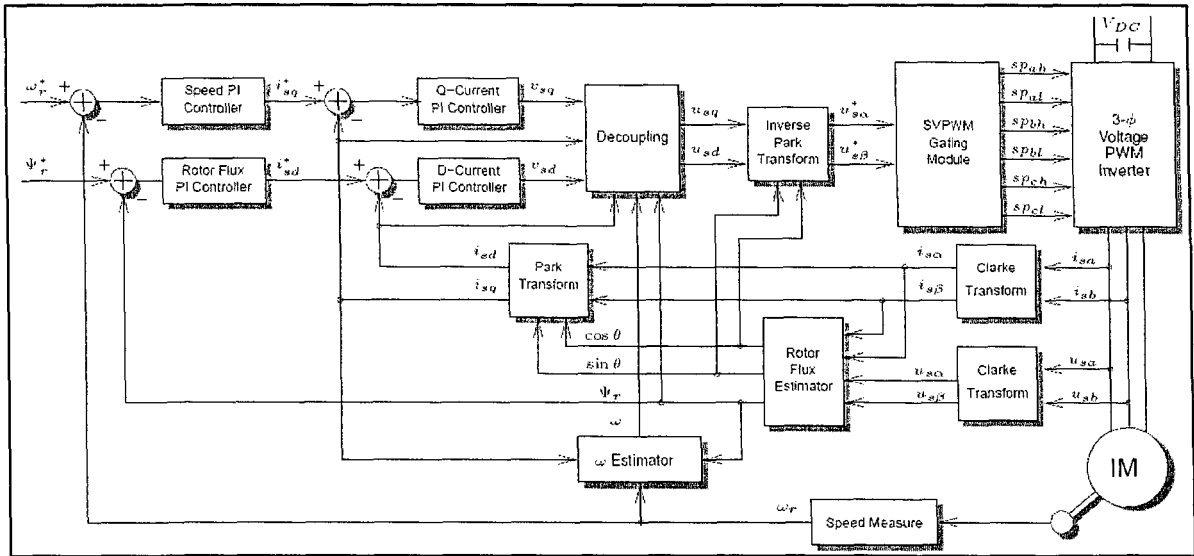


figure 1 - Contrôle vectoriel

La figure 1 montre comment le contrôle vectoriel est scindé sous forme modulaire. Les équations de chaque module se retrouvent à l'annexe A. Dans cette recherche, la lecture de la vitesse se fait avec capteur.

2.1.2 Survol du portillonnage (délais d'activation et temps de maintien)

Avant la venue des larges FPGAs disponibles aujourd'hui, les méthodes utilisées, incluant l'utilisation de DSPs seuls ou en combinaison avec des microcontrôleurs spécialisés, permettaient un temps de cycle total de plus de 100 μ s. Il n'y avait alors pas de manière efficace d'implémenter le SVPWM, ce qui causait l'obtention d'une période de commutation aux alentours de 1-5 kHz, produisant ainsi du bruit audible.

L'introduction de FPGAs plus puissants durant les années 90 a permis pour la première fois de diminuer la quantité de traitement à gérer par le DSP à l'aide de coprocesseurs PWM sur un FPGA de petite taille. Dans le domaine du contrôle de moteur, cela a permis d'augmenter la fréquence de commutation à 10-15 kHz, réduisant nettement le bruit perceptible.

Avec les FPGAs d'aujourd'hui, un contrôle large et complexe comprenant le générateur SVPWM au complet peut se trouver sur une seule puce, tout en supportant des fréquences de l'ordre de 50 kHz. Il serait même possible d'obtenir des fréquences aussi élevées que 100 kHz (transistors MOSFET), mais l'utilité d'une telle vitesse demeure pour l'instant questionnable vu que la dissipation d'énergie est proportionnelle à la fréquence de commutation et que le courant généré par les transistors de puissance (IGBT, GTO) diminue selon la même proportion. Même à 20 kHz, la dissipation d'énergie est importante.

Par exemple, l'équipement utilisé dans notre laboratoire supporte, selon le manufacturier, une fréquence de commutation maximale de 20 kHz. Cependant, il est recommandé de ne pas utiliser une fréquence au-delà de 16 kHz sauf si absolument nécessaire, comme dans le cas d'un contrôle nécessitant une très large bande. Une fréquence de 16 kHz offre l'avantage d'une tension de sortie avec moins de distorsions causées par l'insertion du temps mort. Les simulations du contrôle vectoriel en boucle fermée de cette recherche sont donc paramétrées afin de reproduire une telle fréquence.

Ces désavantages limitent l'intérêt d'utiliser des onduleurs à très haute fréquence de commutation. Un exemple d'application où la fréquence doit être limitée est l'opération de gros moteurs industriels lorsqu'un fort courant est requis de l'onduleur. Pour des applications sur plateformes autonomes (ex. : robots mobiles), c'est la dissipation d'énergie qui doit être minimisée. Une fréquence de 16 kHz semble donc largement suffisante pour la plupart des applications de contrôle, tout en demeurant à l'extérieur de la bande de fréquence audible.

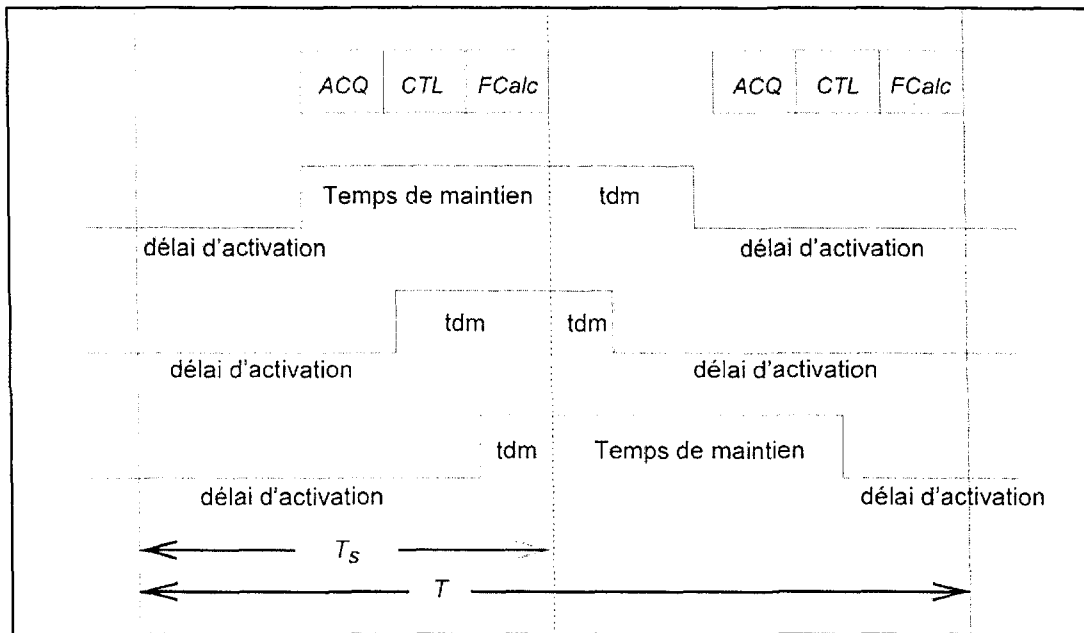


figure 2 - Portionnement PWM asymétrique

Ces contraintes sont prises en considération lors de la maximisation de la fréquence d'échantillonnage et de contrôle. Afin d'optimiser cette dernière sans augmenter la fréquence de commutation, un portionnement PWM asymétrique est employé. Ceci permet d'obtenir une période d'échantillonnage T_s qui est égale à la moitié de la fréquence de commutation T . La figure 2 illustre le cycle d'acquisition (ACQ), de contrôle (CTL) et de calcul des délais d'activation (FCalc) qui sont observables à la fin de la période T_s . Le délai d'activation T_f représente le temps nécessaire avant d'allumer les transistors de puissance, alors que le temps de maintien T_d représente le temps où ces transistors continuent de conduire. Ces temps suivent l'équation $T_f + T_d = T_s$.

Même s'il existe sur le marché des transistors de puissance qui atteignent presque 100 kHz (MOSFET), les calculs de ce travail seront effectués en considérant une fréquence de commutation de 16 kHz exigée par le module de puissance IGBT utilisé dans cette recherche. Cette fréquence offre un temps de commutation de 62.5 μ s, soit un échantillonnage de 31.25 μ s. Il est important de s'assurer d'avoir suffisamment de temps pour l'achèvement du cycle de temps T_{cycle} qui se représente selon l'équation $T_{cycle} = T_{acq} + T_{ctl} + T_{Fcalc}$. La section 3.3 illustre que le

$T_{ctl} + T_{Fcalc}$ nécessite environ $2 \mu s$. Avec des convertisseurs analogues numériques (ADC) qui supportent 1 million d'échantillons par seconde (MSPS), soit $T_{acq} = 1 \mu s$, cela donne un $T_{cycle} = 3 \mu s$, ce qui est très rapide.

Ceci veut dire que la technologie permet de réduire le temps de calcul d'un algorithme complexe de contrôle à quelques μs . Des algorithmes divers pourront donc être modifiés et adaptés afin de bénéficier de cette accélération. Un exemple est illustré à la figure 2 où l'utilisation d'un générateur SVPWM asymétrique permet d'effectuer deux cycles de contrôle complets au lieu d'un seul à l'intérieur d'une période de commutation tout en respectant la physique.

2.1.3 Lecture de vitesse provenant d'un encodeur optique

Le moteur utilisé comporte un encodeur optique afin de fournir l'information concernant la vitesse de rotation actuelle de l'engin. Il y a 5 sorties observées, soit : A, B, INDEX, +5 et GND. Chaque signal correspond à un fil de couleur, illustré à la figure 3.

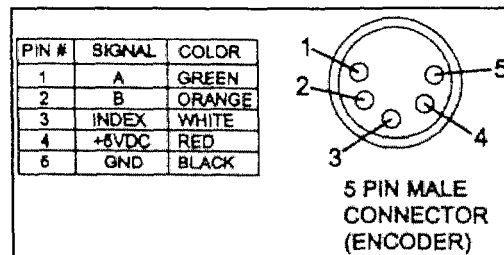


figure 3 - Fils de l'encodeur optique

L'encodeur optique traduit la position angulaire en signal électrique. Pour chaque rotation du moteur, 1000 lignes d'encodage produisent 1000 changements d'état sur les lignes A et B. Donc, plus la période d'un signal sur A ou B est longue, plus le moteur tourne lentement. Le retard entre les signaux A et B permet quant à lui de déterminer la direction de rotation de l'engin.

En cherchant dans la littérature, on trouve l'article [13] qui propose une méthode de lecture de signaux d'un encodeur optique optimale sur FPGA. Les lectures de vitesses sont alors précises pour un grand nombre de scénarios vu la possibilité d'ajuster la méthode de calcul selon la fréquence à traiter. En basse vitesse, l'implémentation utilise une méthode de comptage de période alors qu'en haute vitesse, c'est un comptage de fréquence.

Dans le cadre de ce travail, un simple décodeur sera employé afin de pouvoir plus rapidement passer à la phase de tests. Ce décodeur ne va pas changer, qu'il opère en basse ou en haute vitesse. Vu la présence de 1000 changements d'état par rotation, un intervalle de 1 ms est observé par le FPGA, et la vitesse moyenne est estimée en calculant la valeur de rotation par minute (RPM) correspondante à l'aide de l'équation $X \times \frac{60 s}{1000 \text{ lignes}} = Y \text{ rpm}$ où X est le nombre de lignes détecté par seconde.

Cette méthode de lecture moyenne agit comme un petit filtre sur la vitesse mesurée. Un échantillon de 1 ms est suffisant pour la dynamique du moteur qui sera testé en laboratoire, alors que le filtrage devrait empêcher les vibrations de trop affecter les résultats.

2.2 Référence Matlab de l'algorithme

Même en connaissant les équations qui forment l'algorithme, il est nécessaire de vérifier la validité du prototype. Il faut établir une référence Matlab de l'algorithme; une version fonctionnelle de l'algorithme de contrôle dont les résultats peuvent être accumulés dans le logiciel Matlab. Comme les données numériques sont faciles à importer dans l'environnement Matlab, nos données de référence peuvent être générées dans n'importe quel logiciel qui peut exporter ses résultats. Souvent, cette référence sera le résultat d'une autre recherche ou de l'aboutissement d'autres travaux. Dans le cas présent, un modèle de commande vectoriel développé sous Matlab par le professeur Rachid Beguenane [14] sera utilisé comme référence. Le fait d'établir une telle référence a une conséquence importante pour la réalisation des objectifs : si le prototype donne, avec les mêmes entrées que pour la référence, des sorties qui approchent

celles de la référence en respectant une contrainte d'erreur déterminée, le prototype peut être optimisé et co-simulé. En d'autres termes, le prototype est considéré comme étant valide.

2.2.1 Contrôle Vectoriel

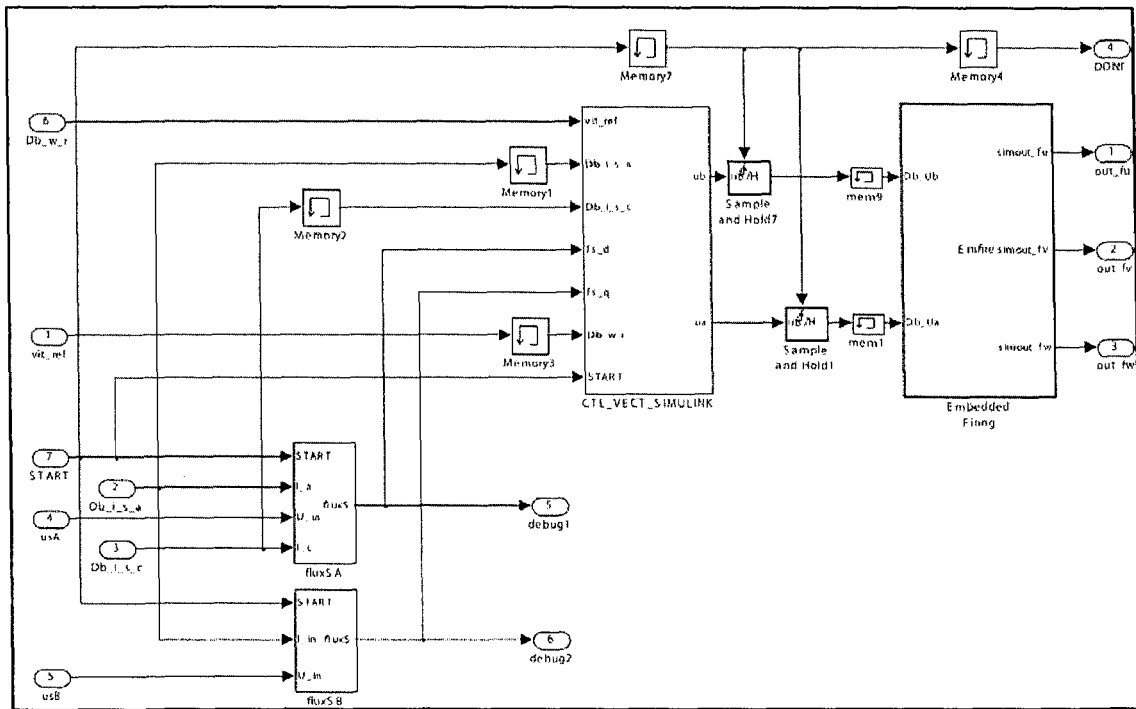


figure 4 - Interface du contrôle Simulink (référence)

La figure 4 représente l'implémentation de référence Simulink du contrôle vectoriel basé sur le modèle de [15]. Comme il s'agit d'une référence, la précision maximale de Matlab est utilisée. L'interface à la sortie et à l'entrée du bloc qui contient le modèle correspond exactement à l'interface de contrôle vectoriel implémenté en utilisant l'outil XSG. Ceci permet de facilement substituer ces systèmes dans l'environnement de test qui comprend les engins SPS ainsi que le bloc de portillonnage (voir section 4.9).

2.2.2 Portionnement (délai d'activation et temps de maintien)

Pour le calcul des délais d'activation ainsi que pour le portionnement, la référence Matlab est facilement transférable vers un bloc de fonction embarquée (code à l'annexe C dans l'environnement Simulink. Ces blocs (figure 5) sont utilisés dans la simulation en boucle fermée de la référence Simulink couplée aux équipements modélisés par SPS. De plus, le bloc portionnement sera réutilisé lors de la co-simulation du contrôle vectoriel XSG car il s'agit d'un processus séparé qui ne peut être co-simulé dans le cadre de cette application (voir explication à la section 4.8).

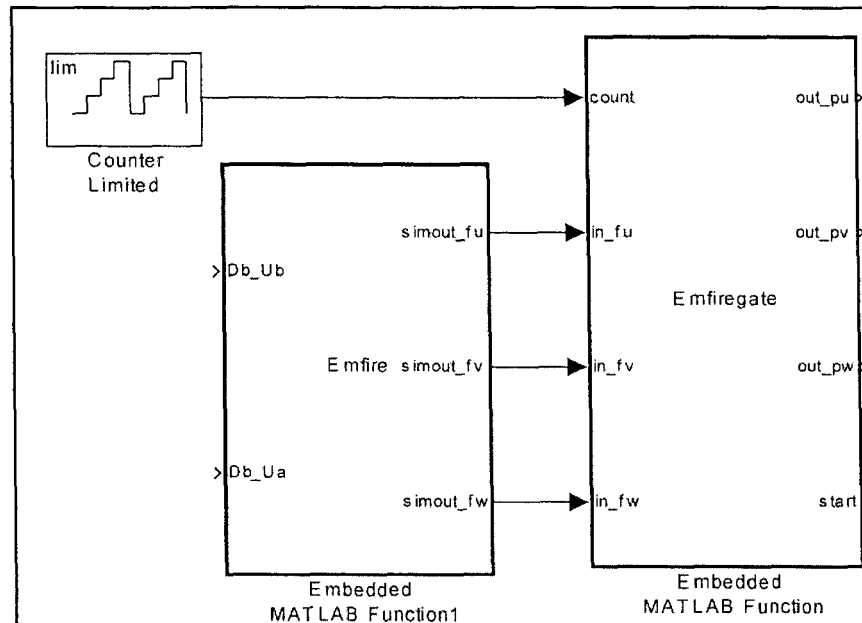


figure 5 - Calcul des délais d'activation et portionnement (embedded Matlab functions)

2.2.3 Moteur et électronique de puissance

Le toolbox SPS est utilisé afin de modéliser le moteur et le module de puissance dans un environnement de contrôle vectoriel complet. Le développement du modèle XSG de contrôle vectoriel dépend fortement des paramètres utilisés pour identifier le moteur SPS simulé. C'est

pourquoi les paramètres de ce moteur partagent la nomenclature des variables utilisées pour le modèle XSG. Il n'y a donc pas de fichier d'initialisation particulier pour le bloc SPS illustré à la figure 6.

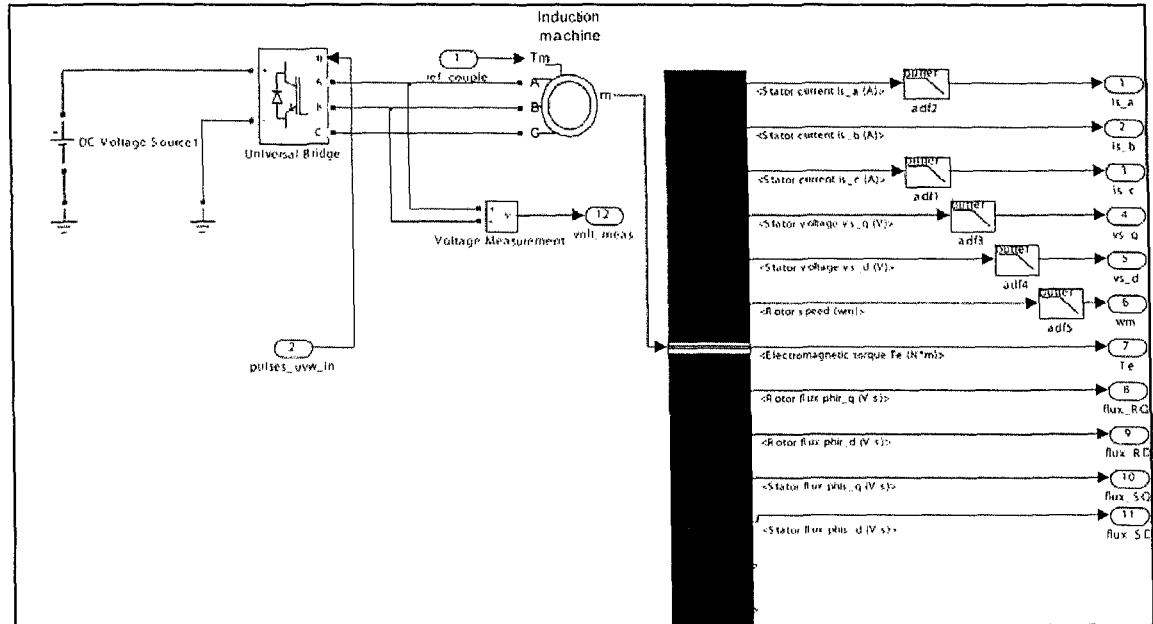


figure 6 - Module de puissance et moteur à induction (SPS)

Lors du développement, un premier ensemble de paramètres est utilisé qui correspond au moteur retrouvé dans un tutoriel de SPS. Ce moteur est choisi pour faciliter le test du portillonnage en comparant les résultats avec ceux du tutoriel SPS de SVPWM. Une fois le modèle XSG fonctionnel, les paramètres du moteur disponible en laboratoire qui sera la cible de tests physiques sont utilisés. Ces paramètres sont illustrés dans le tableau 1.

Moteur Motorsoft 2928	
Puissance	250 W
Vitesse	4000 RPM
Voltage	42 V
P_p	1
R_s	8.1 Ω
L_s, L_R	0.8155 H
R_R	3.8048 Ω
J	0.0014 kg m ²

tableau 1 - Paramètres du moteur en laboratoire

Un des objectifs de la plateforme de développement établie dans cette recherche étant la versatilité du modèle généré, le changement de paramètres entre deux moteurs si complètement différents démontre la flexibilité offerte par la plateforme de travail. Cela démontre aussi que l'optimisation n'est pas trop agressive, ce qui limiterait l'étendue des valeurs que peuvent prendre les paramètres du moteur.

Les tensions et les courants sont filtrés avant leur introduction au contrôle vectoriel. Lors du développement du modèle, des filtres d'ordre 4 et de fréquence de 5000 Hz sont utilisés afin de faciliter le débogage du modèle et de mieux évaluer la précision de sortie de celui-ci. Une fois le modèle assez performant selon les standards du développement, des filtres de fréquence de 500 Hz sont utilisés, ce qui correspond à ceux fabriqués par le technicien de département pour le montage en laboratoire.

2.3 Collecte des données de la référence

La dernière étape préparatoire en ce qui concerne l'algorithme consiste à effectuer la collecte des données de la référence. Comme le prototype XSG est développé bloc par bloc, ces derniers

sont nécessairement reliés par des signaux intermédiaires. Ceux-ci sont bien identifiés lors de la schématisation de l'algorithme, mais ils peuvent être difficiles à identifier dans le modèle de référence. En effet, ce dernier peut être un code complexe sans forme graphique qui se prête mal à l'identification rapide de signaux. Il faut malgré cela isoler chacun de ces signaux (si possible) dans la référence et accumuler les résultats pendant une simulation (figure 7). Ces données seront utilisées pour tester individuellement chaque bloc du prototype avec XSG. Vu la complexité de l'algorithme, il est clair que la quantité de données produites sera énorme. Heureusement, il n'est pas nécessaire d'accumuler une trop grande plage de données pour les signaux intermédiaires. Il s'agit ici de vérifier que nos équations reflètent le comportement de la référence, sans plus. Les cas extrêmes (valeurs trop grandes, trop petites, dépassements, etc.) seront gérés plus tard dans le projet à l'aide de l'outil GAPP (voir section 4.2). Cependant, il faut être plus rigoureux pour l'ensemble du système. Il faut simuler plus longtemps en accumulant les entrées et sorties de l'ensemble de l'algorithme, sans les signaux intermédiaires (figure 8). Dans le cas de la commande vectorielle, un profil de vitesse et de charge est utilisé pour tester le contrôle. La vitesse de référence est changée, ainsi que la charge, et les résultats sont notés. Ceci présente l'avantage de pouvoir afficher la réponse du système sous forme de graphiques qui peuvent ensuite être comparés avec ceux du prototype.

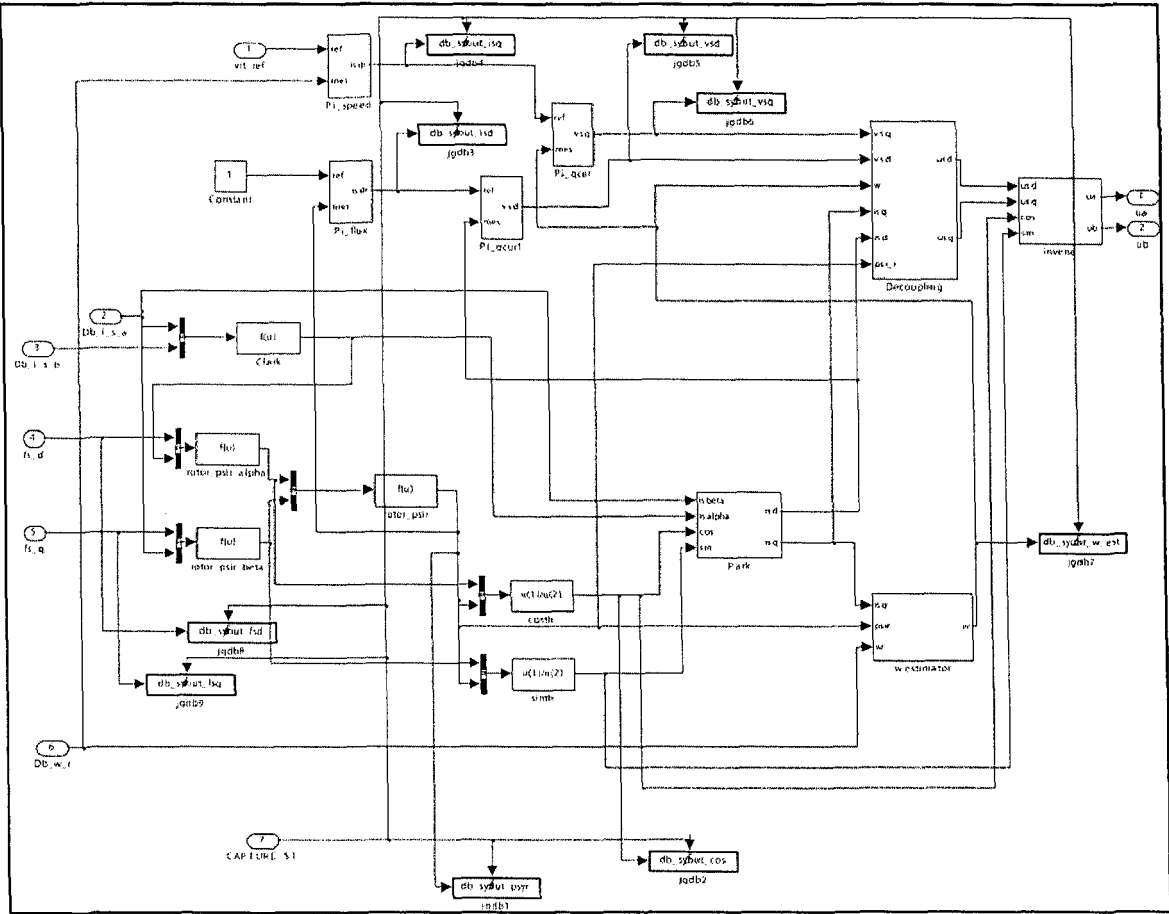


figure 7 - Collecte des données de référence (Contrôle Simulink)

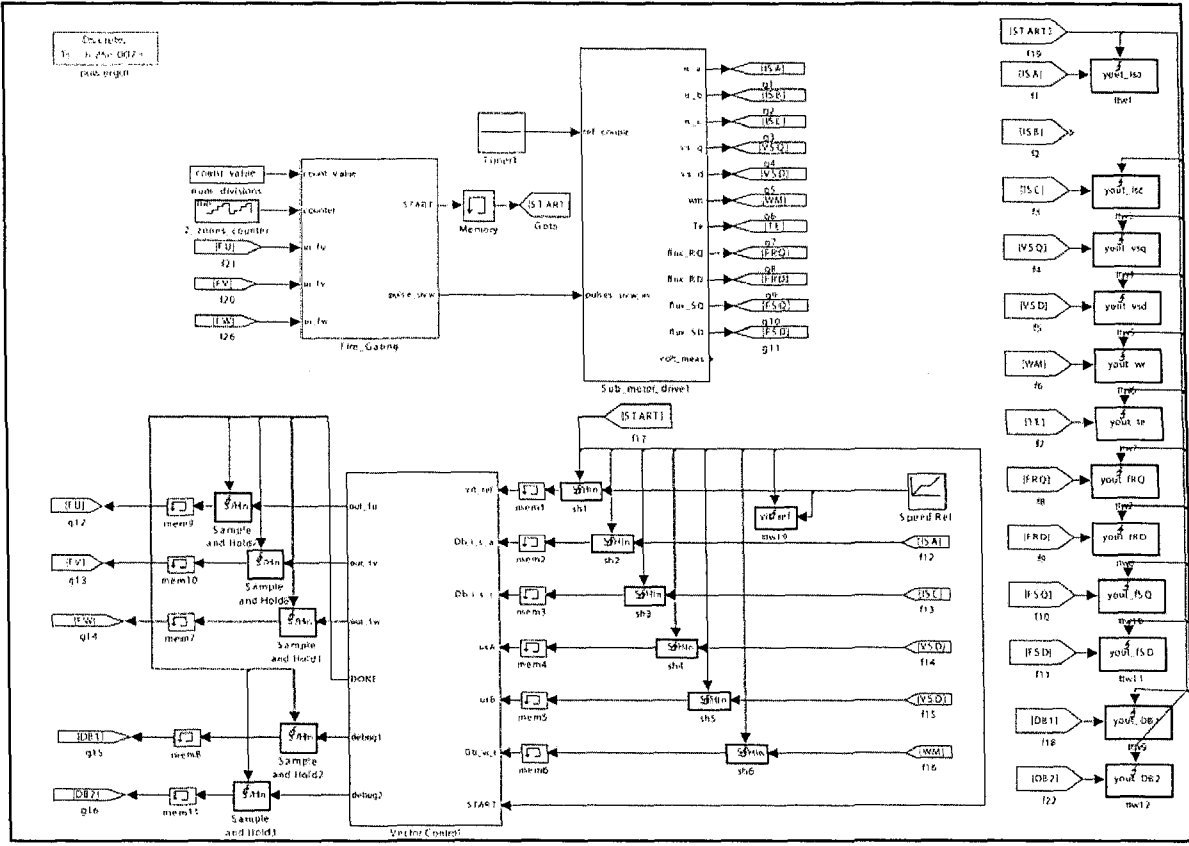


figure 8 - Collecte des données de référence (réponse du moteur)

2.4 Plateforme Amirix

Bien qu'il s'agisse de la dernière étape de développement qui sera suggérée dans la méthode de prototypage rapide produite dans ce travail, il est nécessaire de tout de suite explorer les détails de la plateforme de test en laboratoire. Cette dernière est composée de deux cartes, soit une carte de développement Amirix AP1000 et une carte d'acquisition analogique numérique PMC66-16AISS8AO4 de General Standards. Cette dernière se branche à la carte Amirix via le bus PCI mezzanine (PMC). La recherche mentionne aussi une carte Xilinx ML402. Cette carte est utilisée strictement pour la co-simulation car sa configuration pour cette étape est plus rapide.

Notre achat de carte AIO a influencé CMC qui cherchait une carte d'acquisition à joindre à la plateforme Amirix AP1000 afin d'offrir une solution complète aux groupes de recherche académiques de son réseau. Comme la documentation et les exemples fournis par Amirix ne contiennent pas de projets utilisant la communication via PMC (ni sur la plateforme Linux embarquée, ni sur le *Baseline* de la carte), un contrat entre CMC et l'UQAC (travail réalisé par notre équipe de recherche) fut signé avec pour objectif de produire le nécessaire pour assurer la communication des deux cartes, ainsi qu'une méthodologie simple pour utiliser la carte AIO à partir de la logique matérielle du FPGA. Cet ajout au *Baseline* de la carte Amirix est maintenant utilisé à l'échelle nationale dans la recherche des circuits intégrés et le domaine de MEMS (Micro Electro Mechanical Systems). Comme il s'agit d'une étape de la méthode de prototypage de cette recherche, les détails de développement de la communication sur la carte Amirix sont présentés dans l'annexe B.