

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

**Université Abou Bekr Belkaïd - Tlemcen**  
**Faculté de Technologie**



**Département du Génie Electrique et Electronique**

**Domaine : Sciences et Technologies**

**Master : 2<sup>ème</sup> Année - Semestre 1**

**Filière : Electrotechnique**

**Travaux pratiques**  
**Implémentation d'une commande**  
**numérique en temps réel**

**Dr. Abdelkader Mechernene**

[mechernene\\_aek@hotmail.com](mailto:mechernene_aek@hotmail.com)

**Année universitaire 2025 / 2026**



**Master électrotechnique**  
**Commandes électriques**  
**Niveau : Master 2 – Code matière : EC942**

**Travaux pratiques**

**Implémentation d'une commande numérique en  
temps réel**

**Enseignant : Dr. Abdelkader Mechernene**

**Salle : Laboratoire de Commande des machines électriques**

L'objectif de ce cours consiste à implanter une commande numérique en temps réel, tout en s'initiant au prototypage rapide et au contrôle en temps réel à travers l'utilisation du système dSPACE et du logiciel ControlDesk. La maîtrise de l'utilisation de ce système est primordiale pour la conception, l'implantation et la validation d'applications de commande dans divers domaines du génie électrique, particulièrement celui de l'électrotechnique et de la commande des machines tournantes.

**Année universitaire : 2025-2026**

# Préface

Le présent polycopié de travaux pratiques est consacré à la matière intitulée « *Implémentation d'une commande numérique en temps réel* », destinée aux étudiants de Master 2 de la filière Électrotechnique, spécialité « Commandes électriques », du Département de GEE de la Faculté de Technologie, de l'Université Abou-Bekr Belkaïd.

La commande numérique temps réel représente aujourd'hui un champ incontournable de l'ingénierie électrique. Elle constitue le socle des applications industrielles contemporaines, allant des entraînements électriques intelligents aux systèmes automatisés complexes. Ce polycopié vise à aider les étudiants à transformer leur compréhension théorique des algorithmes de commande en une mise en œuvre pratique sur des plateformes matérielles et logicielles appropriées.

Ainsi, l'étudiant est amené à confronter ses connaissances aux réalités expérimentales, à développer une rigueur méthodologique et à acquérir une autonomie dans la conception et la validation de solutions techniques.

Cette matière s'inscrit dans une logique de formation avancée, visant à doter les étudiants des compétences pratiques indispensables à la maîtrise des systèmes de commande modernes. Les séances de travaux pratiques, d'un volume horaire de trois heures par quinzaine, et les activités proposées sont conçues pour favoriser une démarche scientifique et progressive.

En plus de développer des compétences techniques, les travaux pratiques proposés visent à développer chez l'étudiant des qualités essentielles telles que la précision, l'esprit d'analyse, la capacité de résolution de problèmes et l'aptitude à relier théorie et pratique.

Ces compétences sont au cœur de la formation d'un ingénieur électrotechnicien apte à satisfaire les exigences de l'industrie moderne et à anticiper les évolutions et avancées technologiques. Enfin, ce polycopié se veut un outil pédagogique structuré et accessible, conçu pour accompagner les étudiants dans leur parcours académique et pour enrichir leur expérience pratique.

Nous espérons que ce document contribuera à renforcer l'intérêt des étudiants pour la commande numérique en temps réel et qu'il constituera une référence utile dans leur cheminement vers l'excellence académique et professionnelle.

Dr. Abdelkader Mechernene



# Sommaire

## **TP 01** **(01 semaine)**

### **Prise en main de la carte de prototypage dSPACE 1104 et du logiciel ControlDesk Next Generation**

1. Objectifs
2. Configuration minimale requise par le système dSPACE DS1104
3. Présentation de la carte dSPACE DS1104
4. Conception et mise en œuvre d'un modèle destiné-temps réel dans l'application Simulink
5. Création d'un fichier d'expérience sous ControlDesk Next Generation
6. Compte-rendu

## **TP 02** **(01 semaine)**

### **Mise en œuvre des convertisseurs numérique/analogique et analogique/numérique**

1. Objectifs
2. Matériel nécessaire
3. Mise en œuvre des convertisseurs analogiques/numériques (ADC)
4. Compte-rendu

## **TP 03**

### **Mise en œuvre des capteurs de courant, de tension et de position / vitesse Démarrage d'un moteur asynchrone - Démarrage MCC avec hacheur**

1. Objectifs
2. Matériel nécessaire
3. Démarrage d'un moteur asynchrone triphasé
4. Démarrage direct d'un moteur à courant continu
5. Compte-rendu

## **TP 04** **(02 semaines)**

### **Implémentation des techniques MLI pour onduleur triphasé et hacheur Démarrage MAS avec onduleur- Démarrage MCC avec hacheur**

1. Objectifs
2. Matériel nécessaire
3. Utilisation du convertisseur SEMIKRON Semiteach
4. Démarrage d'un moteur asynchrone triphasé associé à un onduleur
5. Démarrage d'un moteur à courant continu associé à un hacheur 4 quadrants
6. Compte-rendu

**TP 05****(02 semaines)****Implémentation de la commande d'un moteur à courant continu****Commande en couple et en vitesse**

1. Objectifs
2. Matériel nécessaire
3. Présentation du dispositif expérimental
4. Commande en couple du moteur à courant continu
5. Commande en vitesse du moteur à courant continu
6. Compte-rendu

**Annexes**

- A.1 Caractéristiques de la carte dSPACE DS1104
- A.2 Sauvegarde de données
- A.3 Description de l'onduleur multifonctionnel SEMIKTRON
- A.4 Sonde de tension différentielle MX 9030
- A.5 Pince de courant AC et DC i30s
- A.6 Capteur de courant à effet Hall LEM LA55-P
- A.7 Encodeur incrémental
- A.8 Moteurs électriques employés

# Modalités

Une séance de 3 heures par quinze jours et par binôme, encadrée par un (ou deux) enseignant(s) du département de Génie Electrique et Electronique (GEE).

- ✓ Les sous-groupes devront produire un rapport électronique rendant compte du travail réalisé, en y intégrant l'ensemble des résultats obtenus, qu'il s'agisse de calculs, représentations graphiques, données numériques et commentaires.
- ✓ Le compte-rendu sera envoyé à l'adresse indiquée par votre enseignant deux semaines max après la séance (Aucun délai supplémentaire ne sera accordé).
- Rendre votre travail, sous forme électronique, dans une archive nommée : **(TP0X\_Rapport\_Gr\_SGr\_Initiales.rar)** qui contiendra<sup>1</sup> :
  - le fichier **(TP0X\_Rapport\_Gr\_SGr\_Initiales.pdf)** de votre compte-rendu de TP comportant vos résultats (observations, interprétations, conclusions, ...etc.)
  - le dossier de travail **(TP0X\_Gr\_SGr\_Initiales)** contenant les modèles Simulink temps réel, le dossier du projet ControlDesk et l'ensemble des résultats collectés le fichiers **(.mat)**.

## Conseils pour la rédaction des comptes-rendus

Il est possible (et conseillé) de compléter et d'approfondir l'étude du TP par un travail personnel. L'évaluation de ce TP tient compte du compte-rendu et du travail effectué lors de la séance du TP.

Des courbes "brutes" sans commentaire ou des résultats non significatifs peuvent justifier à eux seuls une note de TP très inférieure à la moyenne. Le compte-rendu attendu ne saurait se limiter à une simple liste de résultats. Il doit prendre la forme d'un rapport technique structuré, intégrant des analyses et des observations relatives aux données obtenues. L'attention portée à la pertinence des commentaires ainsi qu'à la qualité de la rédaction constituera un critère essentiel d'évaluation.

Soyez synthétiques et clairs : vous rédigez un rapport, sa lecture doit être facile et les informations importantes doivent être extraites sans difficultés. Les remarques pertinentes sont particulièrement appréciées. Soignez la présentation de vos comptes-rendus : la vue d'une copie mal présentée donne un mauvais à priori. Il sera tenu compte de ces différentes remarques pour la notation.

L'enseignant est en général particulièrement surpris lorsqu'il détecte des informations intéressantes qui ne sont pas la simple reproduction des explications qu'il vous a données. Il est dans ce cas souvent généreux pour la notation. Faites-en sorte que votre copie soit unique, dans le bon sens du terme.

---

<sup>1</sup> **Gr** : Groupe ; **SGr** : Sous-groupe ; **Initiales** : Initiales du rédacteur du compte-rendu ; **0X** : Numéro du TP

# TP 01 : Prise en main de la carte de prototypage dSPACE 1104 et du logiciel ControlDesk Next Generation

## 1. Objectifs

L'objectif de ce TP consiste à :

- apprendre à utiliser la carte de prototypage<sup>2</sup> dSPACE DS1104 en association avec l'environnement logiciel Matlab/Simulink<sup>3</sup> pour établir un modèle dédié-temps réel dans l'application Simulink et produire automatiquement le code en langage C,
- utiliser la librairie (bibliothèque) RTI afin d'interagir avec les composants de la carte DS1104.
- se familiariser avec l'environnement ControlDesk NG pour afficher les courbes temporelles des variables, changer la valeur des variables du programme et les afficher.

## 2. Configuration minimale requise par le système dSPACE DS1104

Pour mettre en œuvre la carte de prototypage dSPACE la configuration minimale est [1], [2], [3] :

- Processeur : Pentium 4 at 2 GHz (ou équivalent et plus),
- Mémoire vive : 2 GB RAM,
- Espace disque : 5,5 GB (et un emplacement dédié)
- Un slot PCI libre 33 MHz/ 2bits 5V sur la carte mère du PC hôte,
- Un port USB pour installer la licence Dongle (Dongle License).

## 3. Organisation de la carte dSPACE DS1104

La commande en temps réel des systèmes continus est assurée par un ordinateur relié à la carte dSPACE DS1104. La conception et la mise en œuvre des programmes s'effectuent au moyen de l'environnement graphique Simulink qui offre une approche graphique permettant de modéliser le problème par l'interconnexion de blocs interconnectés [4].

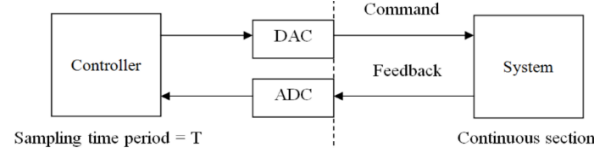


Figure 1. Structure de contrôle en temps réel

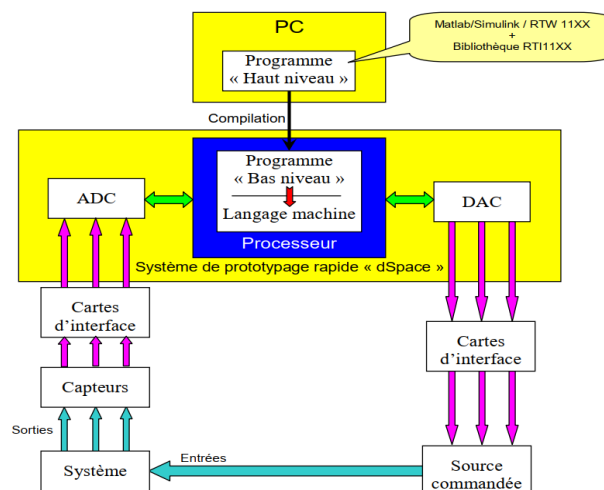


Figure 2. Synoptique d'un système de prototypage rapide

<sup>2</sup> Etape de conception consistant à créer une version préliminaire d'un produit ou d'un système afin de tester ses fonctionnalités et valider ses choix techniques.

<sup>3</sup> Logiciel développé par MathWorks combinant un langage de calcul numérique (MATLAB) et une plateforme graphique de modélisation et de simulation (Simulink). Ensemble, ils permettent de concevoir, analyser et tester des systèmes dynamiques complexes dans divers domaines.

Actuellement, de nombreux environnements de développement temps réel basés sur des processeurs DSP intègrent une interface avec Simulink. Celle-ci permet de transformer directement les modèles graphiques en code exécutable sur une architecture DSP, ce qui réduit fortement le délai nécessaire au à la conception, au développement et au prototypage des solutions de contrôle [1].

Deux méthodes permettent de créer une application temps réel [2] :

- recourir à Simulink pour modéliser le modèle-temps réel et générer automatiquement le code en langage C, puis le compiler et transférer le code compilé dans la mémoire du DSP ;
- programmer directement le modèle-temps réel en langage C, puis le compiler et transférer le code compilé dans la mémoire du DSP.

La première méthode convient aux systèmes simples, sans exigence particulière, elle est la plus efficace car elle accélère fortement le processus de développement et de prototypage.

La programmation du DSP via Matlab/Simulink nécessite la création du modèle temps-réel dans l'interface graphique Simulink. La bibliothèque dédiée, la **DS1104 Real Time Library (RTI1104)**, met à disposition l'ensemble des fonctions indispensables. L'exécution de la commande **Build** permet de générer et de compiler le code C et enfin de le transférer directement le code généré vers le DSP [3].

La dernière étape consiste à exécuter le programme en temps réel et à interagir avec le système. A cette fin, dSPACE propose son interface graphique dédiée, ControlDesk, qui offre la possibilité de visualiser, d'ajuster temps réel et d'analyser les variables issues du modèle Simulink.

La plateforme dSPACE a pour vocation principale de faciliter le prototypage de lois et algorithmes de commande, assurer le pilotage des systèmes et réaliser l'acquisition de données en temps réel. Elle établit une connexion directe entre les programmes développés sous Matlab ou Simulink et les entrées/sorties matérielles. L'intérêt majeur de ce système est de permettre la programmation à l'aide d'un langage de haut niveau (proche de l'homme), accessible et compréhensible par l'utilisateur. La compilation traduit ensuite ce langage en instructions de bas niveau (proche de la machine) adaptées à la machine, assurant ainsi la gestion des entrées et sorties physiques.

L'implémentation d'une commande temps réel, en utilisant la carte dSPACE DS1104 et le logiciel Matlab, nécessite [1], [2], [3] :

- la carte de contrôle dSPACE DS1104 ;
- la licence matérielle sous forme de clé USB (Dongle) ;
- la licence logicielle (fichier Licence.dsp) ;
- le fichier Keys.dsp ;
- le panneau de connexion CP1104 ;
- la bibliothèque spécifique **dSPACE RTI1104**, additionnelle à la bibliothèque originale de Simulink, intégrant les fonctions nécessaires au prototypage rapide et au contrôle temps réel ;
- le logiciel ControlDesk nécessaire à la création d'interface utilisateur destinée à la visualisation et l'acquisition de données.



Figure 3. Dongle

Les trois principales étapes à suivre pour l'implantation rapide d'une loi de commande et de contrôle en temps réel sont alors [2] [3] :

- 1. L'implémentation** : Phase de conception du modèle-temps réel de commande, consistant à élaborer la loi de contrôle dans un du modèle-temps réel Simulink en mobilisant la bibliothèque standard et la librairie complémentaire RTI1104.
- 2. La compilation** : Phase de transformation du modèle-temps réel Simulink en code exécutable (langage C) grâce à l'outil Matlab **Real-Time Workshop Toolbox**. Le code généré est ensuite transféré et stocké dans la mémoire de la carte dSPACE.
- 3. Le contrôle en temps réel** : L'exécution du modèle sur la carte DS1104 et sa supervision sont le plus fréquemment réalisées<sup>4</sup>, en employant l'interface utilisateur **ControlDesk**.

<sup>4</sup> Une autre méthode consiste à effectuer l'exécution et le contrôle à partir de Simulink en utilisant **External Mode Control** [3].

La carte DS1104 est équipée d'un **processeur principal** à virgule flottante, le **MPC8240**, cadencé à 250 MHz. Elle intègre une mémoire de 8 Mo en Flash et de 32 Mo en SDRAM [1], [2], [6], [7]. En complément, elle comporte un **DSP esclave**, le **TMS320F240**, dédié à la génération des signaux PWM<sup>5</sup>.



Figure 4. Carte dSPACE DS1104



Figure 5. Panneau de contrôle CP1104 (Connector Panel)

La configuration fonctionnelle de la carte est donnée par les documents du constructeur, comme montré sur la figure 4 [1], [2], [3], [6], [7].

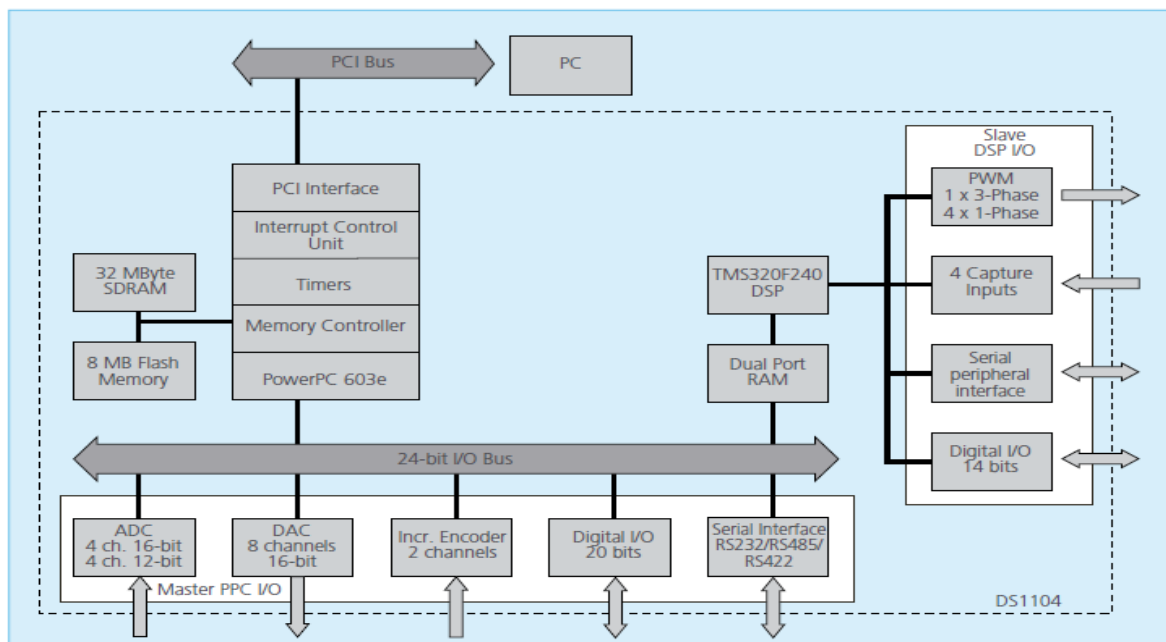


Figure 6. Architecture de la carte dSPACE DS1104

<sup>5</sup> Un signal **PWM, Pulse Width Modulation** en anglais (Modulation de Largeur d'Impulsion ou MLI en français) est une onde périodique dont l'information est codée par la largeur des impulsions. Il permet de contrôler la puissance moyenne transmise à un système en modulant le rapport cyclique entre temps actif et temps inactif.

La carte DS104 est composé de [1], [2], [3], [6], [7] :

- **8 convertisseurs analogiques numériques ADC** : (La tension d'entrée doit être comprise entre -10V et +10V) ;
- **8 convertisseurs numériques analogiques DAC** (La tension délivrée est comprise entre -10V et +10V).

Elle dispose de plusieurs entrées/sorties numériques (Figure 4) [2], [3], [7] :

- **1 interface parallèle 20 bits (20 entrées-sorties numériques)**,
- **2 entrées incrémentales**, (capteurs de vitesse incrémentaux),
- **3 interfaces séries (RS232 / RS422 / RS485)**,
- **3 timers (32 bits)** (utilisables de manière indépendante).

#### Remarque :

Les signaux PWM générés par la carte sont de type TTL, et donc de tension comprise entre 0 et 5 Volts [2], [3].

Les deux types de convertisseurs analogiques / numériques (**ADC**) sont contrôlés par le master PPC.

- **1 convertisseur A/D multiplexé pour les signaux (ADCH1 à ADCH4)** [1], [2], [3], [6], [7] :
  - Résolution : 16 bits ;
  - Plage de tension :  $\pm 10$  V ;
  - Marge d'erreur Offset :  $\pm 5$  mV ;
  - Marge d'erreur sur le gain :  $\pm 0,25\%$  ;
  - Rapport signal sur bruit : SNR >80 dB (à 10 kHz).
- **4 convertisseurs A/D pour les signaux (ADCH5 à ADCH8)** [1], [2], [3], [6], [7] :
  - Résolution : 12 bits ;
  - Plage de tension d'entrée :  $\pm 10$  V ;
  - Marge d'erreur Offset :  $\pm 5$  mV ;
  - Marge d'erreur sur le gain :  $\pm 0,5\%$  ;
  - Rapport signal sur bruit > 70 dB.

Lorsqu'un bloc ADC est intégré dans Simulink, il faut savoir que toutes les entrées analogiques de la carte dSPACE sont automatiquement réduits par un facteur de 1/10, ce qui limite les signaux dans l'intervalle  $-1$  V à  $+1$  V. Pour obtenir la valeur réelle des mesures, **il convient donc de multiplier systématiquement les entrées analogiques par un facteur 10** [1], [2], [3].

Une fois le bloc ADC inséré dans le modèle, il convient de sélectionner le canal d'entrée destiné au signal (numéro de canal).

En effectuant un double-clic sur le bloc, la fenêtre illustrée à la figure 7 apparaît.

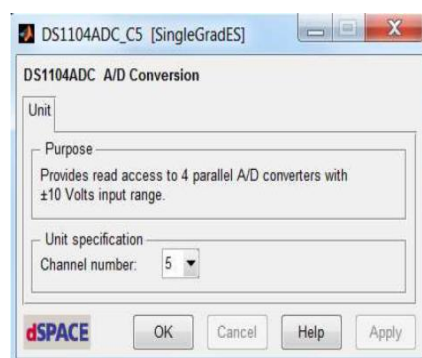


Figure 7

Les caractéristiques des canaux Digital to Analogic (DACH1 à DACH8) sont [1], [2], [3], [6], [7] :

- Résolution 16 bits,
- Plage de tension des sorties analogiques  $\pm 10$  V ( $\pm 5$  mA),
- Plage d'erreur Offset  $\pm 1$  mV,
- Plage d'erreur sur le gain  $\pm 0.1\%$ ,
- Rapport signal sur bruit  $> 80$  dB (à 10 kHz).

De manière identique, l'insertion d'un bloc DAC dans Simulink impose d'appliquer de **diviser par un facteur 10** afin que les signaux restent contenus dans la plage  $-1$  V à  $+1$  V. Il est en outre nécessaire de préciser le numéro du canal de sortie destiné au signal (**Channel number**).

En effectuant un double-clic sur le bloc, la fenêtre représenté sur la figure 8 apparaît.

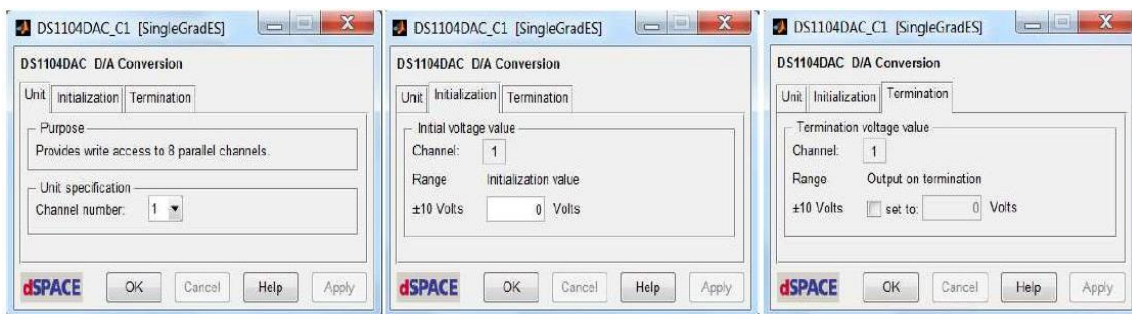


Figure 8

#### 4. Conception et mise en œuvre d'un modèle dédié temps réel dans l'application Simulink

L'exemple suivant consiste à générer un signal à l'aide d'un **Signal Generator**, auquel on additionne une constante, à l'injecter sur la sortie analogique 1 du **Connector Panel** via le convertisseur **DS1104DAC\_C1**. Un oscilloscope relié à la sortie du **Connector Panel** pour visualiser le signal obtenu en sortie.

Pour créer le modèle, il faut auparavant se placer dans un dossier de travail pour construire le fichier Simulink, la construction générant de nombreux fichiers (Voir en annexe).

Créer votre Dossier de travail sur le bureau du PC hôte sous un nom la forme suivante : **TP01\_Gr\_SGr\_Initiales**.

Démarrer le programme Matlab, le message de la figure 9 apparaît, indiquant que dSPACE Real-Time Interface (RTI) est installé pour plusieurs plates-formes matérielles.

Dans notre cas, choisir le modèle de la carte **dSPACE DS1104 (RTI1104)**.

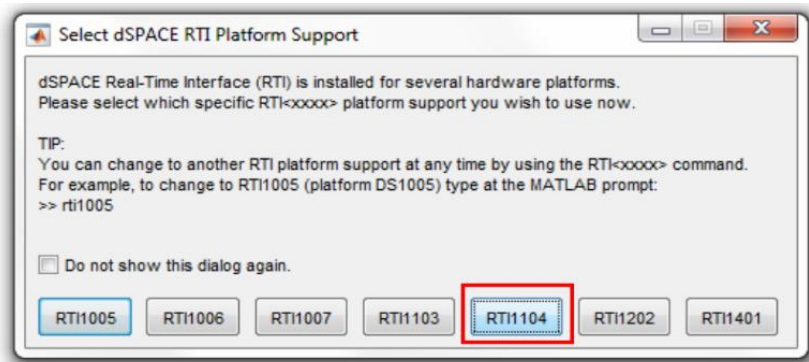


Figure 9

On obtient alors le message suivant [1] :

```

Command Window

-----
Configuring dSPACE(R) Software for MATLAB(R) 8.5.0.197613 (R2015a) ...

RTI           Real-Time Interface to Simulink (RTI1104)  7.4      02-May-2015 okay
RTIFFPGA      RTI FPGA Programming Blockset                       2.9      02-May-2015 okay
DSMPBLIB      Model Port Block Library                             3.0      02-May-2015 okay
DSRT          dSPACE Run-Time Target (DSRT)                       3.0      02-May-2015 okay
DSMSBLIB      Model Separation Block Library                       3.0      02-May-2015 okay
DSMLCON24     dSPACE MATLAB Connection 2.4 (win64)                2.4      01-Jun-2012 okay
-----

*** RTI Platform Support RTI1104 activated.

This is a Classroom License for instructional use only.
Research and commercial use is prohibited.

fx >> |
    
```

Figure 10

Démarrer ensuite l'interface Simulink.

Pour ajouter des blocs de la **librairie RTI1104**, ouvrir la librairie en tapant la commande **RTI1104** (ou à partir de **Simulink Library Browser**) [1].

Cliquer sur **MASTER PPC**, la fenêtre représentée sur la figure 12 s'ouvre.

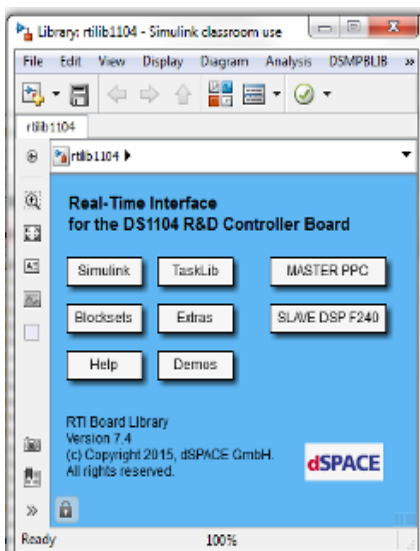


Figure 11

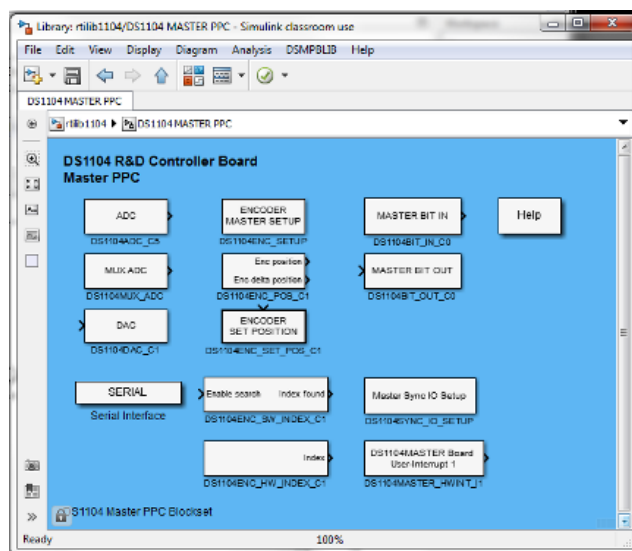


Figure 12

Dans Simulink, choisir **New file** (Figure 13).

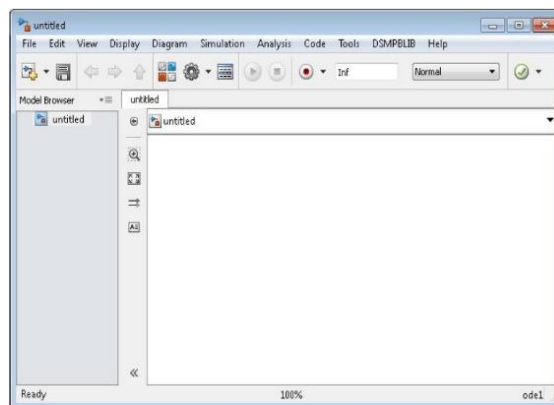


Figure 13

Dans l'interface Simulink, réaliser le modèle représenté sur la figure 14, en ajoutant les blocs fonctionnels à partir de la bibliothèque Simulink et de la **librairie Master PPC** [1].

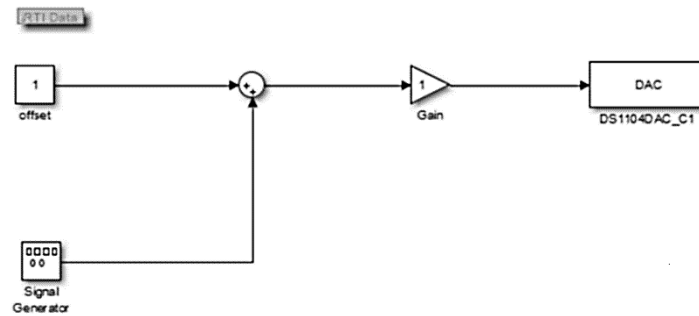


Figure 14

Sauvegarder le fichier dans le dossier de travail sous le nom **test\_01.slx**.

Ensuite, régler les paramètres de simulation comme montré ci-dessous.

- Dans l'onglet **Simulation**, et **Model Configuration Parameters**, puis l'onglet **Solver**, changer le **Stop time** en mettant **Inf** (infini) [1].
- Pour **Solver Options, Type**, choisir **Fixed-step**, puis choisir une période d'échantillonnage (pour cet exemple prendre  $T_e = 100 \mu s$ ) [1], [2], [7].

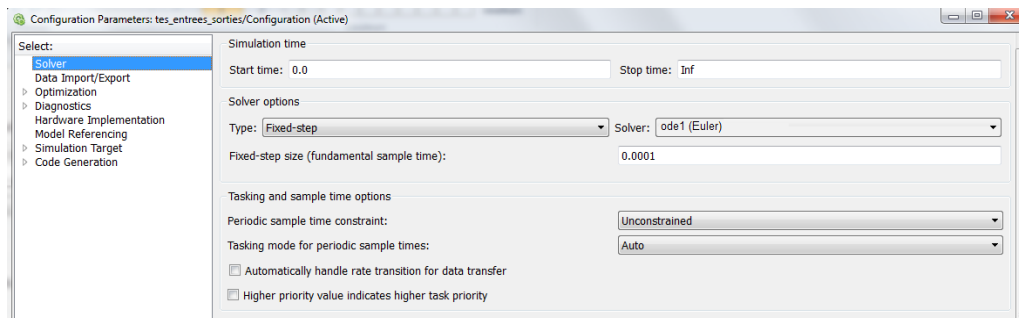


Figure 15

- Dans **Optimization**, décocher la case **Bloc reduction** [1], [2], [7].

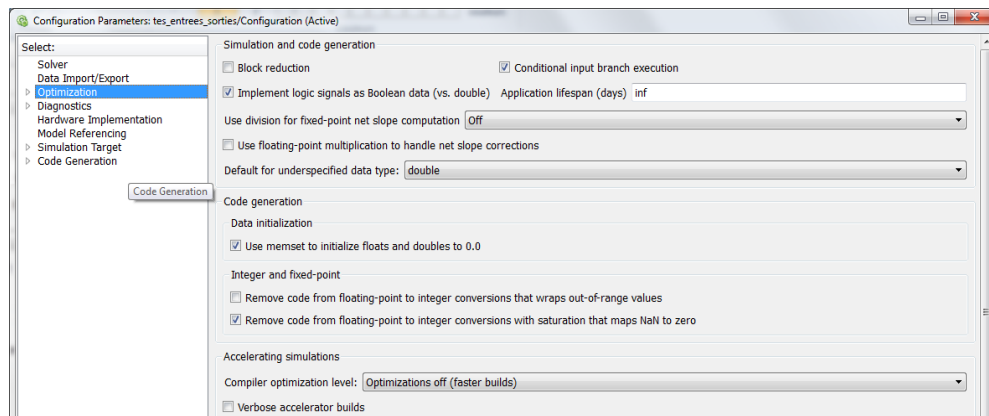


Figure 16

Aller dans **Hardware implementation**, puis **Device Vendor**, choisir : **Generic**, **Device Type** : **Custum**, **Byte ordering** : **Big Endian**, et pour **Signed Integer division rounds to** : **Zero** [1], [2], [7].

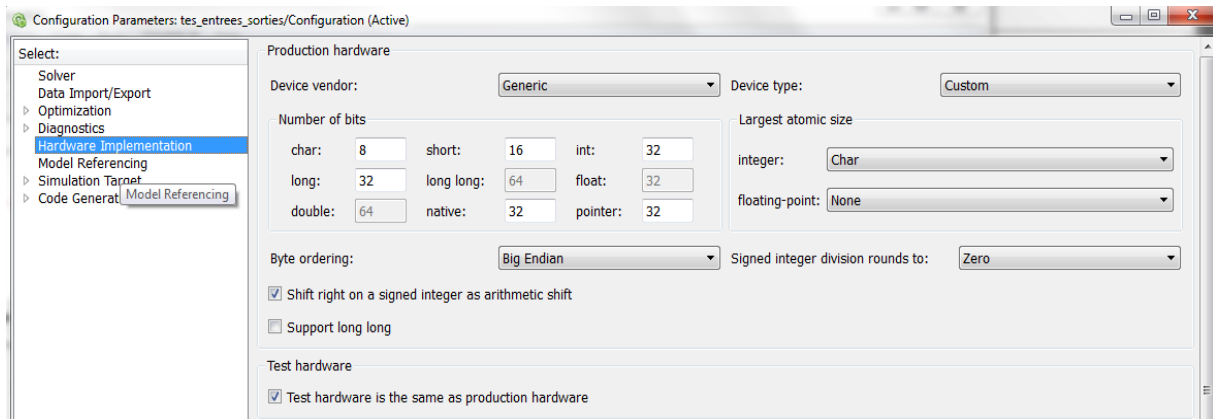


Figure 17

Après, aller dans **Code Generation**, choisir dans **System target file**, le fichier **rti1104.tlc**, et pour **Language**, choisir **C** (afin de générer le code en langage C), puis appuyer sur **Apply** [1], [2], [7].

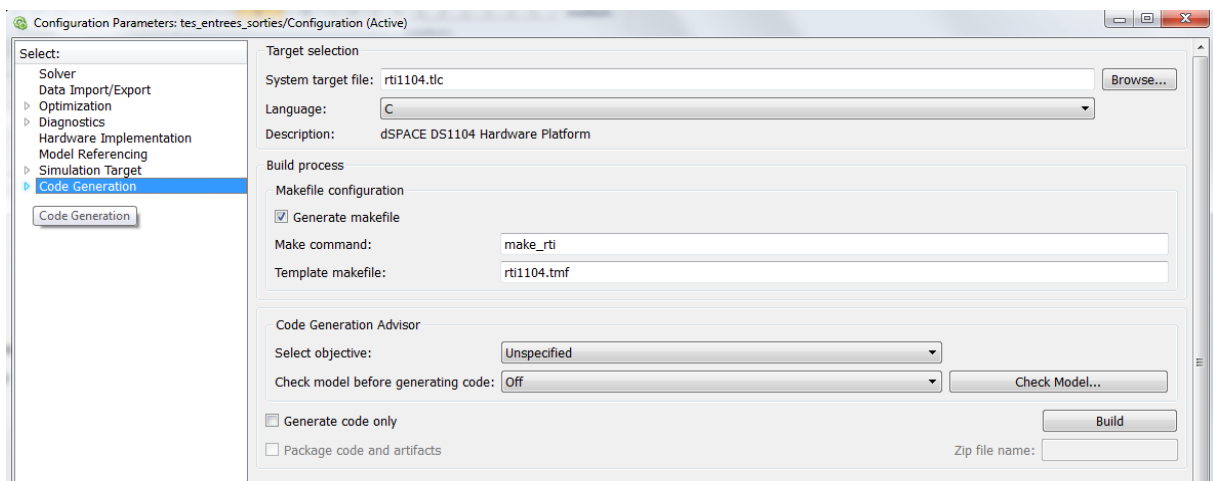



Figure 18

Lorsque le schéma Simulink est terminé, sauvegarder le, et l'étape suivante consiste à générer le code correspondant qui sera sauvegardé automatiquement dans le DSP [1], [2], [7].

La compilation du programme et la génération du code C, est effectué en cliquant sur **control+B**, ou en cliquant sur l'icône  [1], [2], [5].

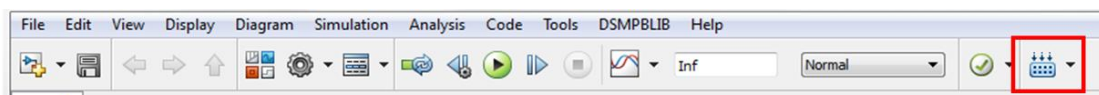


Figure 19

Le code est alors généré, puis compilé et transféré vers la carte dSPACE. Un fichier (**.sdf**) sera généré dans le dossier de travail. Ce fichier comprend toutes les variables utilisées dans Simulink.

Pour vérifier que la procédure de génération de code s'est bien déroulée, vous devrez retrouver le message suivant dans la fenêtre de commande Matlab :

```
### Successful completion of Real-Time Workshop build procedure for model: test_01
*** Finished RTI build procedure for model test_01_tr
```

Désormais, l'étape suivante consiste à aborder le travail en temps réel.

## 5. Création d'un fichier d'expérience sous ControlDesk Next Generation

Pour visualiser et piloter en temps réel les différentes variables du fichier implémenté sous Simulink, on utilise le logiciel ControlDesk NG de la société dSPACE.

ControlDesk est une interface utilisateur (Homme-Machine) qui permet de visualiser en temps réel différentes variables du programme développé sous Simulink et de modifier également des paramètres définissant le mode de fonctionnement des blocs constituant le schéma Simulink [1], [7].

La visualisation de variables ou de signaux par des graphes et la modification de paramètres par des boutons d'action associés au programme Simulink ou à l'acquisition sont possibles par l'intermédiaire d'instruments graphiques que l'on peut sélectionner dans la fenêtre **Instruments** [1], [7].

L'espace de travail sous ControlDesk est composé [1], [7] :


- d'une **fenêtre de navigation (Navigator)**, elle liste le **Layout**, la carte connectée et le programme en cours,
- de l'**aire de travail** contenant le Layout,
- d'une fenêtre **Instrument selector** contenant les instruments pour la création d'un **Layout** et d'une fenêtre d'outils (**Tools window**), la fenêtre des variables Simulink (**fichier.sdf**).

Les variables sont représentées sous forme d'arbre en correspondance avec les blocs et sous-blocs du programme Simulink.

Il existe trois modes de fonctionnement sous ControlDesk [1], [2] [3], [6] [7] :

- **le mode Layouting** : Mode de construction de Layout,
- **le mode Standard** : Mode permettant de tester la validité des connexions entre le Layout et les variables Simulink.
- **le mode Measuring** : Mode d'exécution du programme contenu dans le DSP avec le Layout comme IHM (Interface Homme Machine).

### • Création d'un nouveau projet sous ControlDesk

Double-cliquer sur l'icône ControlDesk NG  à partir du raccourci se trouvant sur le bureau, ou aller dans le menu démarrer de Windows et lancer le programme. La fenêtre suivante s'ouvre [1] :

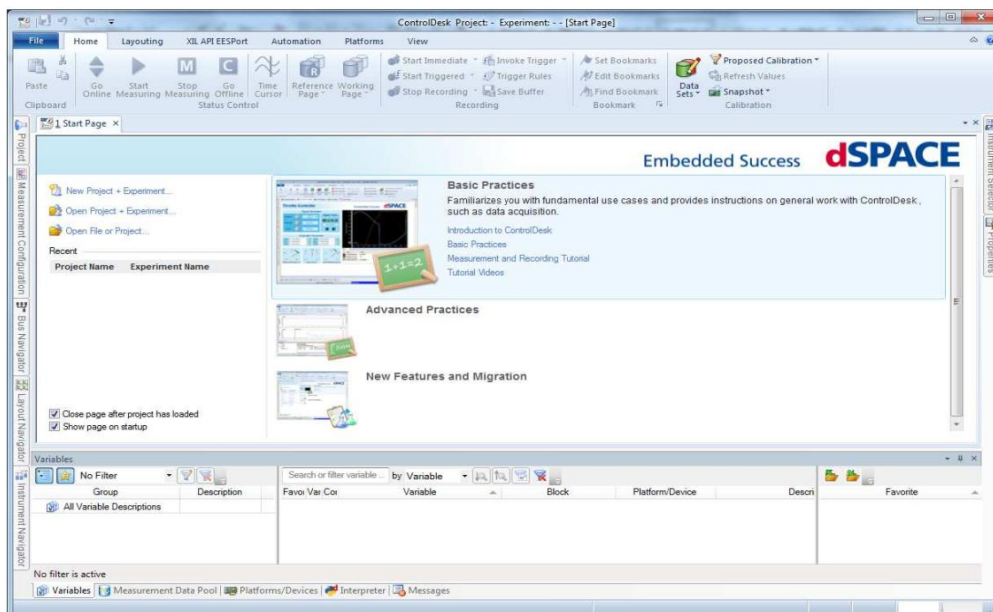


Figure 20

Dans ControlDesk, un projet doit contenir au moins un **Experiment**.

Dans le menu **File**, cliquer sur **New Project + Experiment**, puis saisir un nom (**Name of the project**) et choisir le dossier dans lequel sera enregistré le projet (Dossier de travail déjà créé) [1].

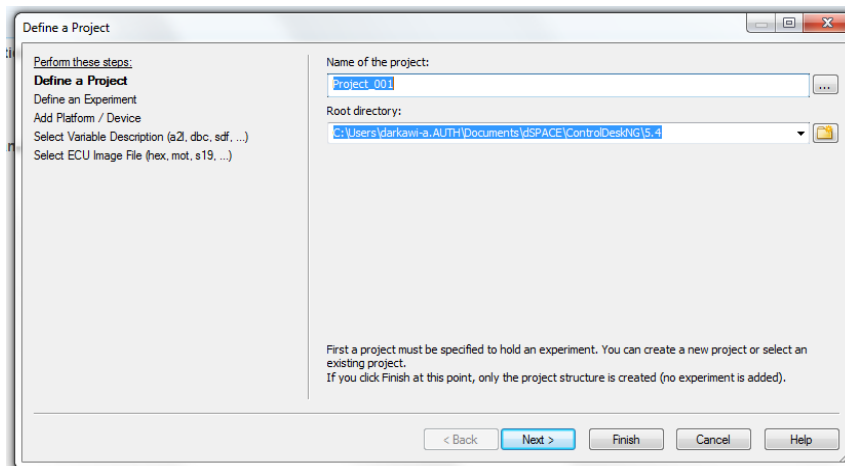


Figure 21

La fenêtre de définition du projet s'ouvre. Saisir le nom du projet, puis cliquer sur suivant (Next).

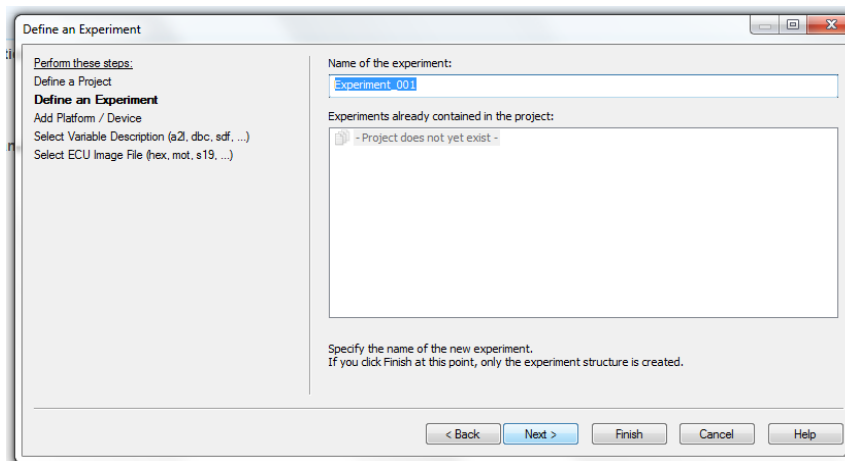


Figure 22

Sélectionner ensuite la plateforme (DS1104), puis cliquer sur Next.

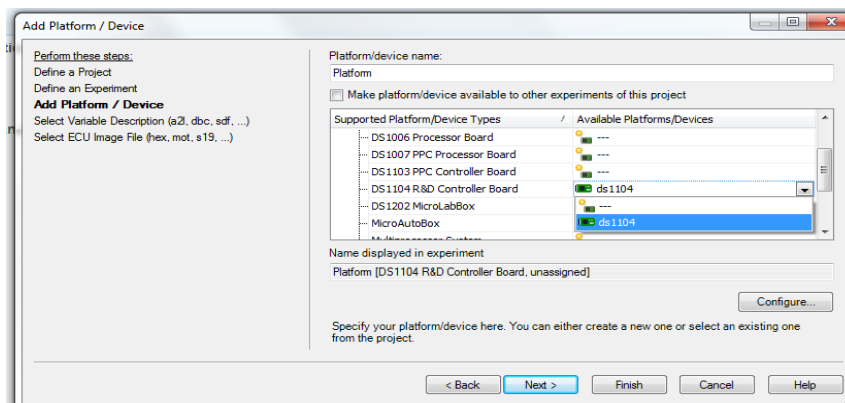


Figure 23

Dans la fenêtre **Select variable description**, choisir le fichier (**.sdf**) correspondant (Fichier généré lors de l'exécution de votre programme Simulink) et cliquer sur **Finish** [1].

Le fichier (**.sdf**) est utilisé par ControlDesk pour lier les variables du fichier Simulink aux variables affichées ou contrôlées dans le Layout.

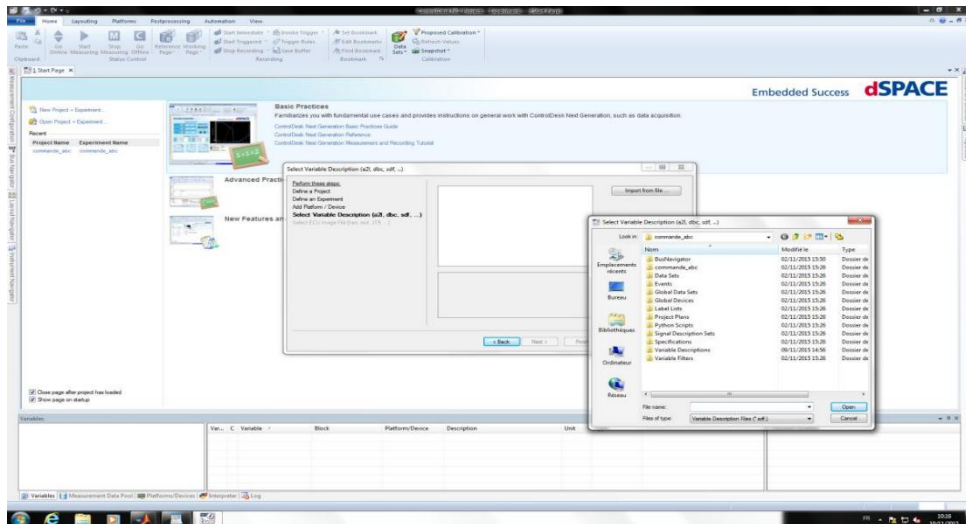


Figure 24

Le projet est créé, il faut maintenant créer un (ou plusieurs) **Layout** associé à ce projet.

Un **Layout** est une interface graphique à laquelle on peut ajouter divers instruments de contrôle et/ou de visualisation, afin de visualiser et/ou modifier en temps réel les différentes variables du projet [1].

Les variables disponibles via Simulink peuvent être vues dans le bas de la fenêtre des variables (Figure 23). La fenêtre (Onglet **Projet**) montre le fichier (.sdf) sélectionné pour le chargement.

**Remarque**

Lors de toute modification au modèle Simulink d'origine, en cliquant sur l'onglet **Projet**, faites un clic droit sur le fichier (sdf), recharger la description de la variable. Vous pouvez mettre à jour les variables à leurs dernières valeurs sans redémarrer dSPACE.

Exemple, on peut changer les paramètres d'un régulateur en temps réel, modifier une consigne, etc...

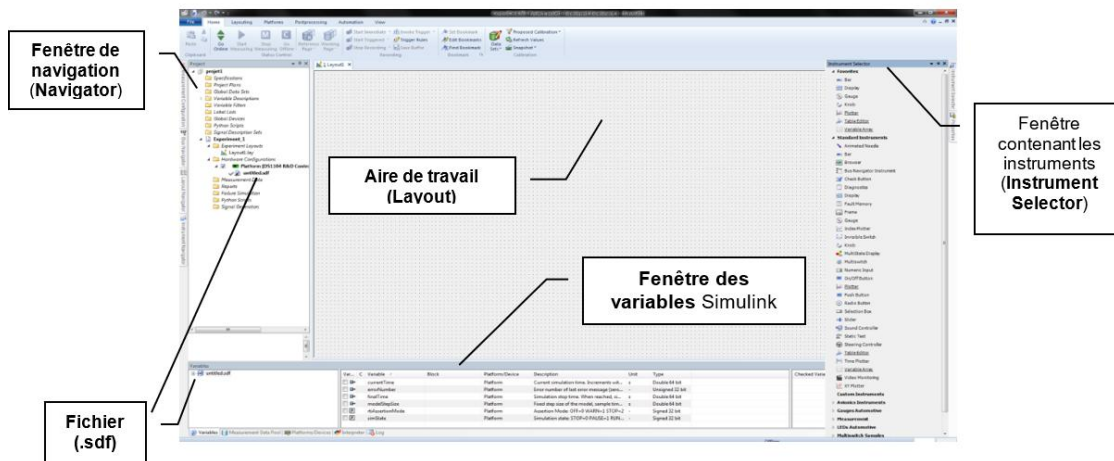


Figure 25

Le layout est construit dans l'espace de travail, les instruments y sont ajoutés depuis la fenêtre **Instrument Selector**. (Si l'une de ces fenêtres n'est plus visible à l'écran, on peut l'ouvrir dans **View** puis **Controlbar**).

Deux types d'instruments sont mis à disposition [1] :

- les instruments de visualisation et/ou de modification de variable : les **Virtual Instruments**,
- les instruments qui permettant l'acquisition de variables (à l'écran ou dans un fichier) : les **Data Instruments**.

ControlDesk permet d'éditer de nouveaux instruments personnalisés, puis de les sauvegarder dans une librairie sous l'onglet **Custom instruments**.

Après avoir rajouté des instruments, il faut affecter à chaque bloc une variable et modifier les propriétés de l'instrument si besoin en cliquant sur le bouton droit et ensuite sur **Properties** [1], [7].

Pour réaliser une interface "Tableau de bord" il suffit d'insérer, à partir de la fenêtre **Instrument Selector**, les objets graphiques nécessaires en les faisant glisser sur le layout.

A titre d'exemple, sur la figure 24, trois instruments virtuels ont été inséré :

- un **Numeric Input**, pour régler un paramètre numériquement au clavier,
- un **Radio Button**, pour effectuer un choix d'entrée,
- un **Slider**, pour régler un paramètre en faisant glisser un curseur.





Figure 262

### • Faire une mesure

Dans cette étape, on affecte des variables du fichier Simulink aux instruments virtuels correspondants en sélectionnant la variable souhaitée, puis en la déplaçant vers l'instrument dans l'espace de travail.

### Remarque

- Le symbole  signifie que l'on ne peut que visualiser la variable [1].
- Tandis que le symbole  signifie que l'on peut visualiser et modifier la variable [1].

Depuis **Instrument selector**, insérer dans l'espace de travail un **Plotter**, et une **Gauge** (pour visualiser le signal de sortie) et deux **Numeric Input** pour modifier la fréquence et l'amplitude du signal.

Ensuite, glisser les variables correspondantes à chaque instrument virtuel.

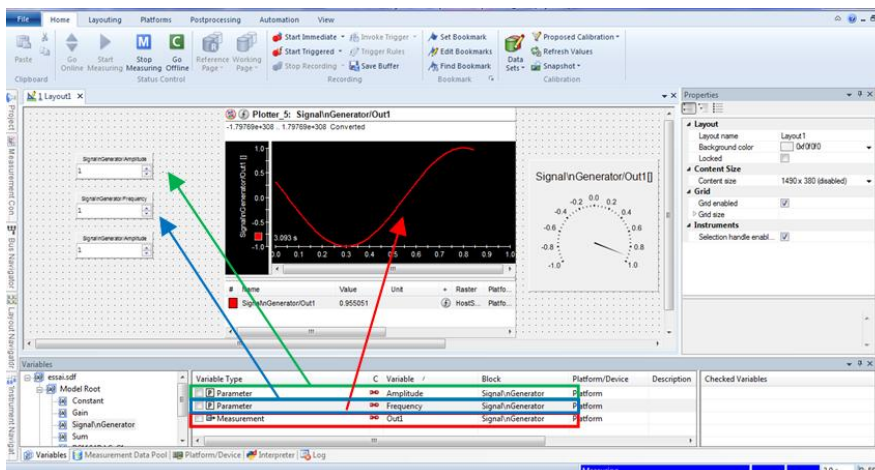


Figure 27

- **Animation mode (Fonctionnement en temps réel)**

Afin de lancer l'exécution du programme, cliquer sur **Go Online** (Figure 28).

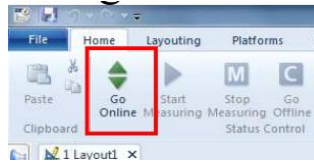


Figure 28

Pour lancer l'exécution du modèle, l'acquisition des mesures et agir sur le programme Simulink, utiliser le mode **Measuring** en cliquant sur le bouton **Start Measuring** (Figure 29).

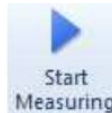


Figure 29

Attention, avant de recompiler (Build) sous Matlab/Simulink, il faut impérativement désactiver le mode **Online** en cliquant sur **Go Offline** (Figure 30).

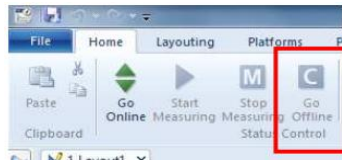


Figure 30

A la fin de la compilation, il sera possible de remettre le mode **Online** pour charger les modifications.

- **Acquisition de données**

ControlDesk permet de réaliser des acquisitions de données.

Pour augmenter la durée de visualisation des mesures, aller dans **Measurement Configuration > Triggers > Duration Trigger1**, puis changer la durée dans **Duration**, (Par exemple 1s) [1], [7].

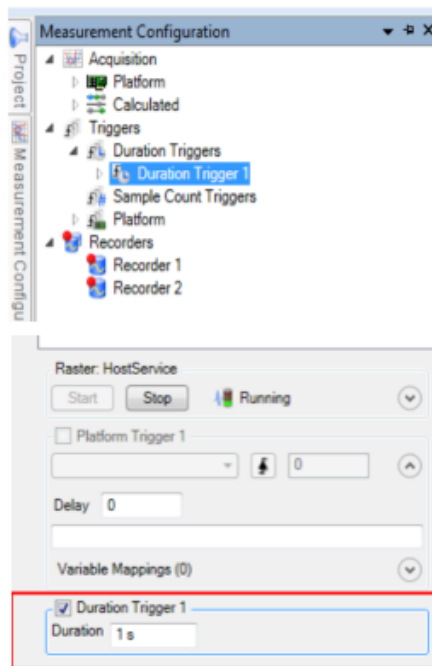


Figure 31

Si on le désire, il est possible de modifier les propriétés des axes X et Y, il suffit alors de cliquer avec le bouton droit, puis sur **Properties**. Dans la fenêtre **Properties**, Axis, dans **Scaling mode** choisir **fixed**, puis remplacer par les valeurs max et min désirées.

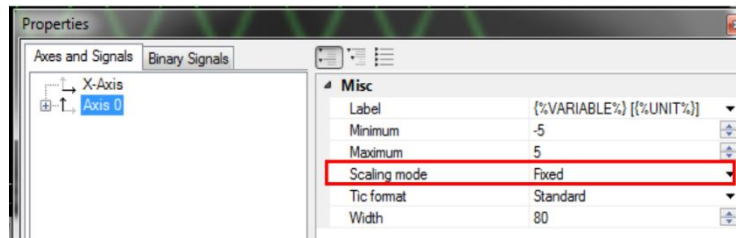


Figure 32

- **Sauvegarder des données**

Pour enregistrer les données (mesures) dans votre ordinateur, aller sur **Measurement** puis **View Measurement Configuration**, puis dans **Recorders**, puis choisir **Create New Recorder** [8].

Nommer l'enregistrement, puis glisser les variables que vous désirez mesurer dans le tableau **Recorder** (Figure 33).

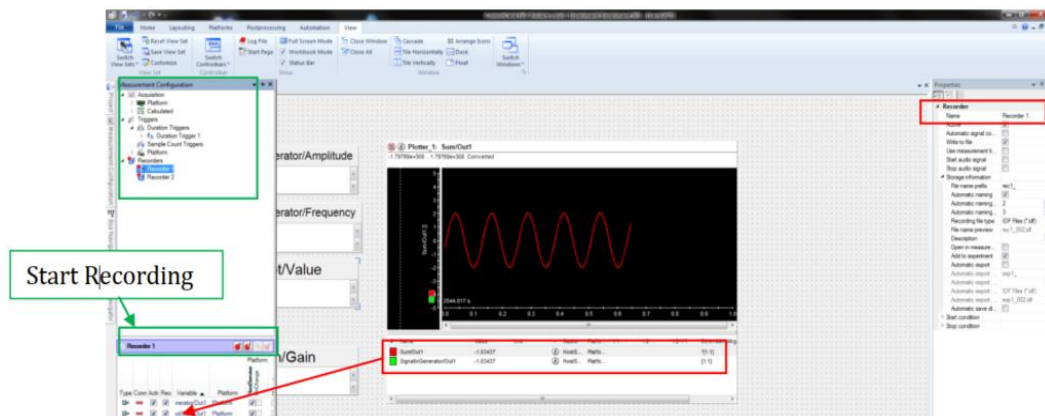


Figure 33

Dans la fenêtre **Properties**, ajuster les paramètres de sauvegarde, puis débiter l'enregistrement en cliquant sur **Star Immediate Recording** :

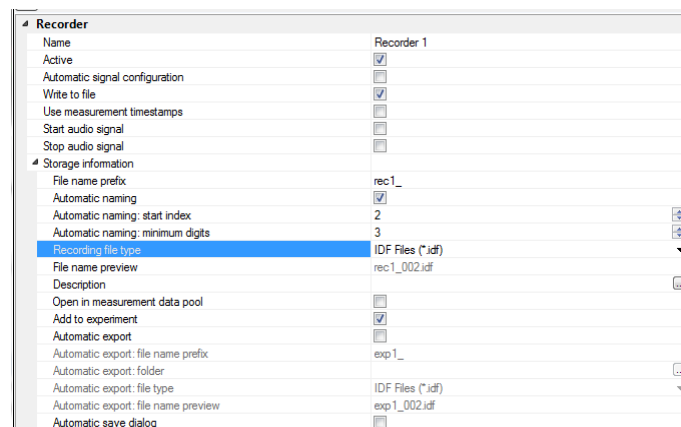


Figure 34

ControlDesk affecte par défaut le nom **rec1.idf** pour le fichier d'enregistrement.

Pour exporter les données vers un fichier Matlab, aller dans **Project, Experiment1, Measurement Data** et faite un clic droit sur le fichier **(.idf)** créé, puis **Export** en choisissant le type de fichier **(.mat)** dans l'onglet **Save as Type** [1].

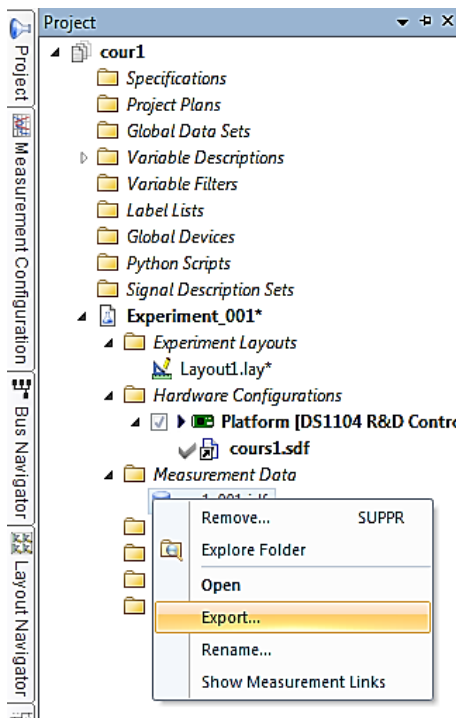


Figure 35

Une fois que vous avez exporté le fichier (.mat) (Nommer le fichier : **test\_01.mat**, sauvegarder le dans le dossier de travail), vous pouvez le charger dans l'espace de travail de Matlab et extraire les variables en utilisant le programme de récupération des données suivant [1], [3], [9].

```

% ***** %
% Script pour la récupération de données sous Matlab
% Nom du fichier : test_01.mat
% Disposer le fichier dans le dossier de travail de Matlab
% ***** %
%
load test_01.mat ;           % Pour charger les données (Fichier : test_01.mat)
%
temps = test_01.X.Data;     % Pour générer le vecteur temps
%
variable_1 = test_01.Y(1).Data; % Récupération des variables
variable_2 = test_01.Y(2).Data;
% ... idem jusqu'à la variable n
variable_n = test_01.Y(n).Data; % Variable numéro n (n : Nombre de variables enregistrées)

% ***** %
% *** Traçage des courbes *** %
% ***** %
figure 1;
plot(temps, variable_1); grid;

figure 2;
plot(temps, variable_2); grid;
...
% ... idem jusqu'à la variable n
figure n;
plot(temps, variable_n); grid;

% Fin
% ***** %

```

## 6. Compte-rendu

Rendre votre travail, sous forme électronique, dans une archive nommée : **(TP01\_Rapport\_Gr\_SGr\_Initiales.rar)** qui contiendra :

- le fichier (**TP01\_Rapport\_Gr\_SGr\_Initiales.pdf**) de votre compte-rendu de TP comportant vos résultats (observations, interprétations, conclusions, ...etc.)
- le dossier de travail (**TP01\_Gr\_SGr\_Initiales**) contenant les modèles Simulink temps réel, le dossier du projet ControlDesk et l'ensemble des résultats collectés le fichiers (**.mat**).

***Fermer les applications ouvertes, éteindre l'ordinateur et couper l'alimentation du poste de travail.***

***Ranger le poste de travail et le remettre le dans l'état dans lequel vous l'avez trouvé en début de séance.***



## TP 02 : Mise en œuvre des convertisseurs numérique/analogique et analogique/numérique

### 1. Objectifs

L'objectif de ce TP consiste à utiliser la carte dSPACE DS1104 et à se familiariser à l'emploi des convertisseurs analogique/numérique (ADC) et numérique/analogique (DAC).

### 2. Matériel nécessaire

- 01 générateur basse fréquence (GBF),
- 01 oscilloscope numérique,
- 01 voltmètre magnétoélectrique ou numérique,
- 01 carte de prototypage dSPACE DS1104 avec panneau de contrôle CP1104,
- 01 ordinateur hôte avec le logiciel Matlab/Simulink, la bibliothèque « dSPACE RT1104 » et l'environnement ControlDesk NG.

### 3. Mise en œuvre des convertisseurs analogiques/numériques (ADC)

On désire, dans un premier temps, générer un signal numérique, puis un second signal analogique fourni par un générateur basse fréquence récupéré sur une entrée analogique via un convertisseur ADC. Les signaux seront appliqués à l'entrée d'un filtre du premier ordre et injectés sur les sorties analogiques via un convertisseur DAC.

#### • Construction du modèle Simulink

- Démarrer le programme Matlab, puis choisir la carte dSPACE DS1104 utilisée (**RT1104**).
- Ouvrir un nouveau fichier Simulink, puis charger la **librairie RT1104** à partir de **Simulink Library Browser** (ou en tapant la commande RT1104).
- Implanter le modèle Simulink en y insérant, depuis la librairie Master **Master PPC**, un convertisseur ADC (**DS1104ADC\_C5**) et un convertisseur DAC (**DS1104DAC\_C1**).
- Créer **votre propre Dossier de travail** sous un nom la forme suivante : **TP02\_Gr\_SGr\_Initiales**, puis sauvegarder le fichier sous le nom **test\_02.slx** dans le fichier de travail.

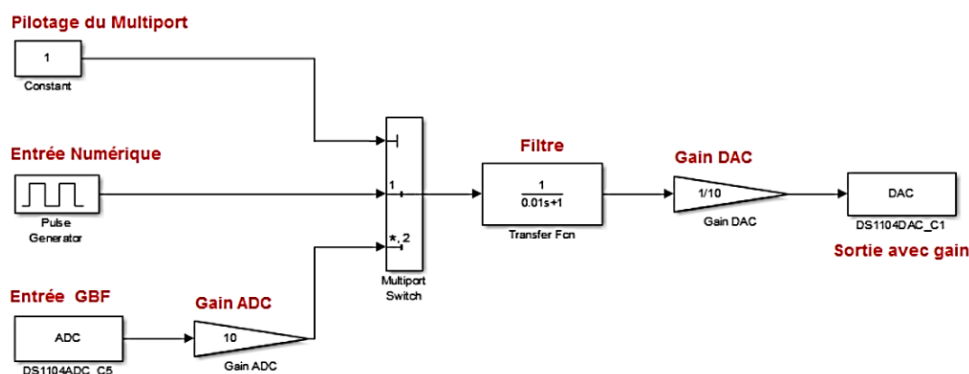



Figure 1

- Pour compiler le programme, cliquez sur l'icône  (ou en tapant [**Control+B**] au clavier).
- Prés la compilation, le code C sera généré et transféré vers la carte dSPACE. Le fichier **test\_02.sdf** sera ensuite généré et placé dans le Dossier de travail (Ce fichier renferme toutes les variables définies par votre programme Simulink).

- Pour vérifier que la procédure de génération de code s'est bien déroulée, vous devrez retrouver le message suivant dans la fenêtre de commande Matlab :

```
### Successful completion of Real-Time Workshop build procedure for model: test_02
*** Finished RTI build procedure for model test_02_tr
```

- L'étape suivante consiste au travail en temps réel.

### 3.1 Implémentation du programme et visualisation des sorties du système avec ControlDesk

- Lancer l'application ControlDesk et sélectionner **New Project + Experiment** dans l'interface d'accueil.
- Saisir un nom (par exemple : **Project\_001**), et indiquer le dossier de travail **TP02\_Gr\_SGr\_Initiales**, dans lequel sera sauvegarder le projet.
- Choisir le fichier (**test\_02.sdf**), en se référant au dossier de travail, puis cliquer sur **Finish**.
- Un onglet supplémentaire s'ouvre dans le cadre inférieur (**Variables**) portant le nom du fichier (**test\_ordre1.sdf**). Dans cet onglet, vous trouverez une arborescence (**Variable browser**) commençant par le nom de votre programme puis en dessous le sous Dossier « Model root ».
- Ouvrir ce niveau, vous devez retrouver les noms des variables de votre programme Simulink. Cela va permettre de créer un lien entre ces variables et l'interface ControlDesk.
- Insérer, depuis l'onglet du menu **Instrument Selector**, les instruments virtuels nécessaires pour réaliser votre interface « Tableau de bord », à titre d'exemple :
  - un bouton glissant **Slider Control**, pour régler les paramètres du générateur PULSE (Amplitude, période, rapport cyclique, déphasage) ;
  - un bouton radio **Radio Button** pour piloter le multiport (choix d'entrée) ;
  - des entrées numériques **Numeric input** pour régler les paramètres du filtre,
  - un **Plotter** pour visualiser les différents signaux.
- Au final, vous obtiendrez l'interface suivante (Figure 2) :

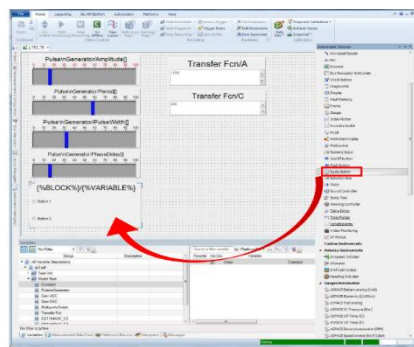


Figure 2

- Affecter les variables du fichier Simulink aux instruments virtuels correspondants, en sélectionnant la variable souhaitée, puis en la déplaçant vers l'instrument dans l'espace de travail.

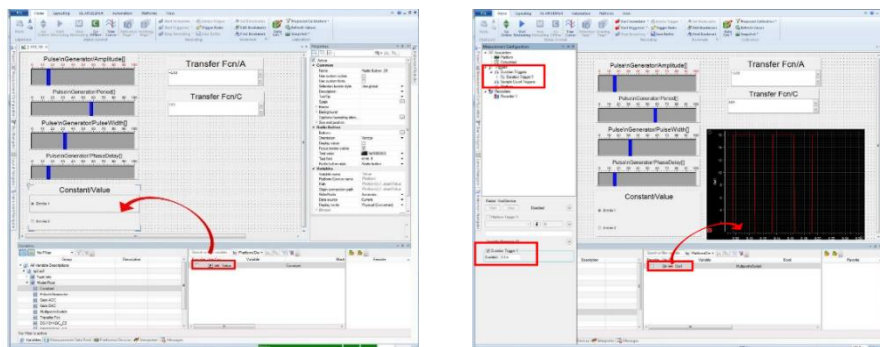


Figure 3

### 3.2 Génération d'un signal numérique (Pulse generator)

- **Mesures**

- En sortie du panneau de contrôle, connecter la sortie DACH1 à un oscilloscope numérique pour visualiser le signal délivré en sortie du filtre.
- Sélectionner l'entrée 1 du commutateur multiport (**Multi-port Switch**).
- Lancer l'exécution du modèle et l'acquisition des mesures en cliquant sur **Start Measuring**.
- Visualiser les signaux d'entrée (**Pulse Generator**) et de sortie du filtre à l'aide du **Plotter** et de l'oscilloscope.
- Modifier les paramètres d'entrée du générateur **Pulse Generator** (Amplitude, période et rapport cyclique) et visualiser les différents signaux.
- Que peut-on conclure ?

- **Acquisition et sauvegarde**

- ✓ **Influence de la période du signal d'entrée Pulse Generator**

- Sélectionner les paramètres suivants :
  - Rapport cyclique : 0,5                      Déphasage : 0
  - Période  $T = 10$  s                      Amplitude = 1 Volt
- Visualiser et sauvegarder, sur un même graphique, les signaux d'entrée et de sortie du filtre.

**Remarque :** La période et le rapport cyclique sont exprimés en [ms] (Interface ControlDesk).

- Analyser la réponse du filtre (Temps de réponse, paramètres, amplitude, qualité du signal ...).
- Pour le signal précédant (Rapport cyclique : 0,5 ; déphasage : 0 et amplitude : 1 Volt), modifier la période ( $T = 0.01$ s) ; visualiser et sauvegarder sur le même graphique les signaux d'entrée et de sortie du filtre.
- Comparer les résultats obtenus pour les deux cas étudiés :  $T = 10$  s et  $T = 0,01$  s.
- Que peut-on observer ? Conclure.

- ✓ **Modification des paramètres du filtre**

- Choisir une valeur de 10 pour le numérateur et de -1 pour le dénominateur. Les paramètres du signal d'entrée sont 1 Volt pour l'amplitude et  $T = 10$  s pour la période.
- Visualiser et sauvegarder les deux signaux (à l'entrée et à la sortie du filtre en utilisant Matlab, à l'aide de l'acquisition de signaux à partir de ControlDesk)
- Que peut-on observer ? Conclure.

### 3.3 Acquisition et modification d'un signal extérieur (Entrée extérieure GBF)

- **Mesures, acquisition et sauvegarde**

- Réaliser à l'aide d'un générateur basse fréquence (GBF) un signal de type **PULSE** ayant pour amplitude 1 Volt et une période  $T = 10$  s.
- Visualiser ce signal sur l'oscilloscope numérique.
- Brancher le signal produit par le GBF, sur l'entrée analogique ADC (**DS1104ADC\_C5**).
- **Attention :** L'affichage des valeurs de tension du GBF sera programmé en considérant que la sortie est à haute impédance. La séquence de programmation est la suivante :  
**Top Menu >>> Output Menu >>> Load Impedance >>> HIGH Z**
- Sélectionner l'entrée 2 du commutateur multiport (**Multi-port Switch**).
- Lancer l'exécution du modèle et l'acquisition des mesures en cliquant sur **Start Measuring**.
- Visualiser et sauvegarder sur le même graphique les signaux d'entrée et de sortie du filtre.

- Analyser la réponse du filtre (Temps de réponse, paramètres, amplitude, qualité du signal ...).
- Comparer les résultats obtenus pour les signaux d'entrée et de sortie du filtre.
- Comparer les signaux en sortie du convertisseur « ADC » et délivré par le générateur externe.
- Quel est le rôle du **Gain\_ADC**, en déduire le rôle du **Gain\_DAC**.
- Modifier les paramètres du signal délivré par le GBF externe (Type, fréquence et amplitude **inférieure ou égale à 10 V**).
- Visualiser et sauvegarder les signaux.
- Conclure.

#### **4. Compte-rendu**

Rendre votre travail, sous forme électronique, dans une archive nommée :

**(TP02\_Rapport\_Gr\_SGr\_Initiales.rar)** qui contiendra :

- le fichier (**TP02\_Rapport\_Gr\_SGr\_Initiales.pdf**) de votre compte-rendu de TP comportant vos résultats (observations, interprétations, conclusions, ...etc.)
- le dossier de travail (**TP02\_Rapport\_Gr\_SGr\_Initiales**) contenant les modèles Simulink temps réel, le projet ControlDesk et l'ensemble des résultats collectés le fichiers (**.mat**).

***Fermer les applications ouvertes, éteindre l'ordinateur et couper l'alimentation du poste de travail.***

***Ranger le poste de travail et le remettre le dans l'état dans lequel vous l'avez trouvé en début de séance.***



## TP 03 : Mise en œuvre des capteurs de courant, de tension et de position/vitesse Démarrage d'un moteur asynchrone - Démarrage d'un moteur à courant continu

### 1. Objectifs

L'objectif de ce TP consiste à :

- apprendre à mettre en œuvre les capteurs de courant, de tension et de vitesse en association avec la carte dSPACE DS1104 ;
- faire l'acquisition et visualiser à l'aide du logiciel ControlDesk les courbes temporelles des variables : courant, tension et vitesse lors des démarrages d'un moteur asynchrone et d'un moteur à courant continu.

### 2. Matériel nécessaire

- 01 alimentation triphasé variable (3 x 0 à 430 Volts AC / 5A),
- 01 machine asynchrone triphasé à cage 400/230 V-50Hz,
- 01 machine à courant continu,
- 02 capteurs de courant (sondes Fluke i30s) de rapport 100 mV/A, pour la mesure des courants  $i_a$  et  $i_b$  circulant dans les phases A et B.
- 01 capteur de tension (sonde différentielle Metrix MX 9030) de rapport 1/20 et 1/200 pour la mesure de la tension  $u_{bc}$  entre les phases A et B du moteur,
- 01 capteur incrémental de vitesse (Encodeur) Baumer-GI355, 1024 points pour la mesure de la vitesse ou de la position de l'arbre du moteur,
- 01 voltmètre, 01 ampèremètre et 01 tachymètre (pour la vérification de la mesure de la vitesse),
- 01 carte de prototypage dSPACE DS1104 avec panneau de contrôle CP1104,
- 01 ordinateur hôte avec le logiciel Matlab/Simulink, la bibliothèque « dSPACE RT1104 » et l'environnement ControlDesk NG.

### 3. Démarrage d'un moteur asynchrone triphasé

#### • Présentation du dispositif expérimental

Le schéma du dispositif expérimental étudié est représenté sur la figure 1.

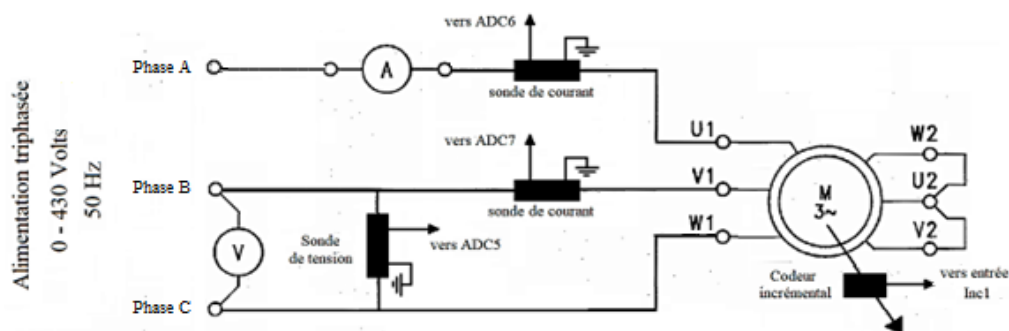


Figure 1

Le rôle des capteurs dans le montage est d'effectuer les mesures, mais aussi de réaliser l'isolation galvanique entre le circuit de puissance du moteur et la carte dSPACE DS1104.

- Les courants  $i_a$  et  $i_b$  circulant dans les phases A et B seront mesurés, le troisième courant  $i_c$  circulant dans la phase C ne sera pas mesuré, mais déduit des deux autres courants à l'aide de la relation :  $i_a + i_b + i_c = 0$ .
- La fréquence d'échantillonnage  $f_e$  choisie sera de 10 kHz ( $T_e = 0,1$  ms) (Il faut que  $f_e \gg 50$  Hz).

Les sorties des différents capteurs seront connectées aux entrées suivantes :

- Tension  $u_{bc}$  : Entrée ADC, canal ADCH5 ; ( $u_{bc}$  : Tension mesurée entre les phases B et C)
- Courant  $i_a$  : Entrée ADC, canal ADCH6 ;
- Courant  $i_b$  : Entrée ADC, canal ADCH7 ;
- Vitesse de rotation : Entrée Inc1, ( $N$  en tr/mn) et  $\Omega$  en rad/s).

### • Mise en œuvre pratique des capteurs de courant et de tension

Pour la mesure de courants et de tension, on utilise respectivement des capteurs (sondes) de courant et de tension (capteur à effet Hall type LEM par exemple). Il est alors nécessaire de déterminer le gain (Rapport) de chacun des capteurs et la valeur de leur offset<sup>6</sup> (décalage) [2], [10], [11].

Le capteur de courant donne en sortie une tension  $v_{mes-i}$  proportionnelle au courant qui le traverse  $i$ , et dont on désire effectuer la mesure (Figure 2.a) [2], [12] :

$$i = G_i \times v_{mes-i}$$

De la même manière, la tension  $v_{mes-v}$  est proportionnelle à la tension  $v$  que l'on désire mesurer aux bornes du capteur (Figure 2.b) :

$$v = G_v \times v_{mes-v}$$

Les gains  $G_i$  et  $G_v$  dépendent de la résistance interne  $R_m$  des capteurs. Cette résistance est dimensionnée pour remettre à l'échelle les deux grandeurs électriques. En effet, les tensions limites des entrées ADC de la carte DS1104 sont  $\pm 10V$ , les capteurs mesurent les grandeurs, mais adapte également les mesures à cette plage [2], [12].

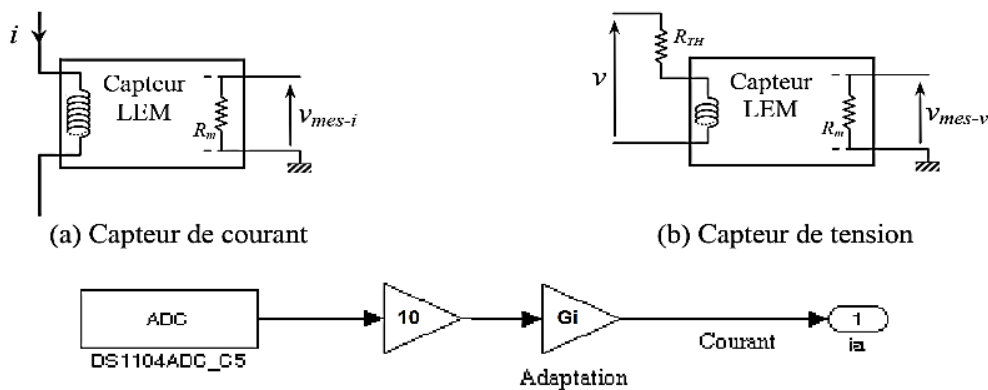


Figure 2. Remise à l'échelle des mesures

Les mesures doivent ensuite être filtrées pour éliminer les bruits de mesure, deux techniques sont couramment mises en œuvre :

- L'emploi d'un filtre passe bas ayant une fréquence de coupure convenable. Ce filtre doit être dimensionné afin d'être transparent pour la plage des fréquences de fonctionnement possibles (par exemple  $\omega_c = 1000 \text{ rad/s} \Rightarrow f_c \approx 160 \text{ rad/s}$ ) [2]. La figure 3 montre un exemple de schéma bloc pour la mesure pour ce cas.

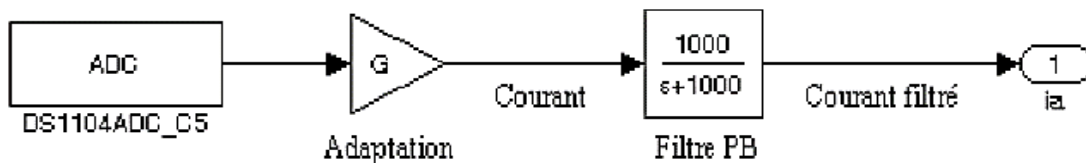


Figure 3. Remise à l'échelle et filtrage des mesures

<sup>6</sup> Présence d'une valeur en sortie du capteur, lorsque la grandeur à mesurée est nulle.

- L'emploi d'un bloc permettant de lisser les mesures par le calcul de la moyenne des  $n$  derniers échantillons. Ce bloc peut être ajouté au modèle à l'aide d'un *sub-system*, comme montré sur la figure 4 [11].

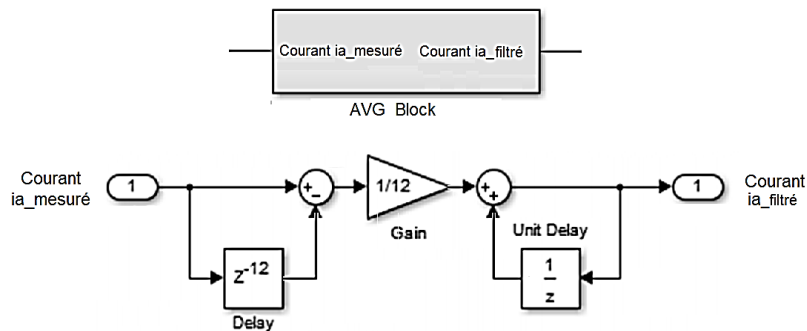


Figure 4. Lissage par calcul de la moyenne des  $n$  (12) derniers échantillons

De plus les capteurs de mesures sont généralement affectés d'offset. Dans les applications où ces défauts ne peuvent être négligés par rapports à l'amplitude de la grandeur mesurée, il faut les déterminer, puis les éliminer avant l'exploitation des mesures [2], [11].

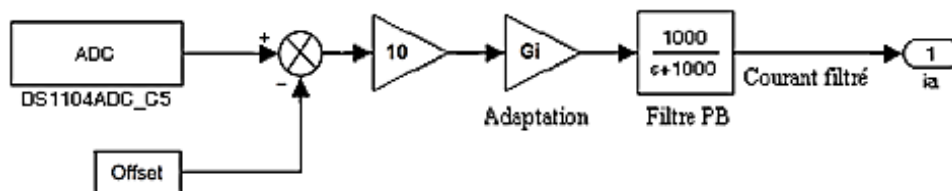


Figure 5. Compensation de l'offset

### • Mise en œuvre pratique d'un encodeur incrémental pour mesure de la vitesse

La vitesse peut être mesurée par une génératrice tachymétrique, relié arbre à l'arbre avec le moteur qui l'entraîne, par une simple mesure de la tension d'induit. Celle-ci est alors appliqué sur une entrée ADC. Une seconde méthode consiste à utiliser un encodeur incrémental. L'encodeur est lié à l'arbre du moteur, le câble est connecté à la carte et produit les signaux **A**, **B** et **Z**, via une entrée incrémentale disponible sur le panneau de contrôle. La mise en marche de l'encodeur est réalisée dans l'environnement Matlab/Simulink grâce à l'interface temps réel **RTI1104**. La visualisation des mesures est faite avec l'interface ControlDesk [2], [12].

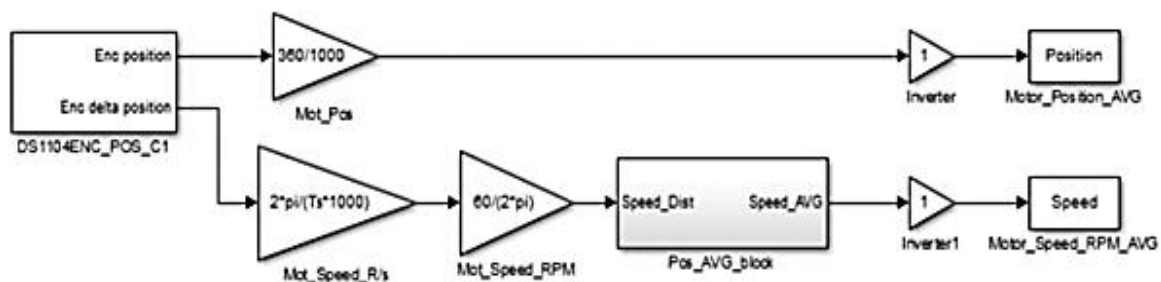


Figure 6. Mesure de position et vitesse de rotation rotor par un codeur incrémental

#### a. Préparation des blocs [2], [11]

- Dans 'Toolbox dSpace RTI1104', ouvrir le menu 'Library : rtilib1104/DS1104 MASTER PPC'.
- Insérer, depuis cette librairie, dans un fichier .slx les blocs suivants :
  - ENCODER MASTER SETUP : **DS1104ENC\_SETUP**
  - ENC position/delta position : **DS1104ENC\_POS\_C1/2**
  - INDEX : **DS1104ENC\_HW\_INDEX\_C1/2**

## b. Configuration des blocs [2], [11]

- **ENCODER MASTER SETUP** : (Figure 7.a)

DS1104ENC\_SETUP

Encoder signal type (type d'entrée) : single-ended (TTL) ou Differential (RS422)

Choisir le type 'single-ended'.

- **ENC position/delta position** :

DS1104ENC\_POS\_C1/2 (Figure 7.b)

Unit : Channel number 1 or 2

Initialization (initial position value) : de -2097152 à +2097152 (221)

Choisir 'Channel 1 and initial position value = 0'.

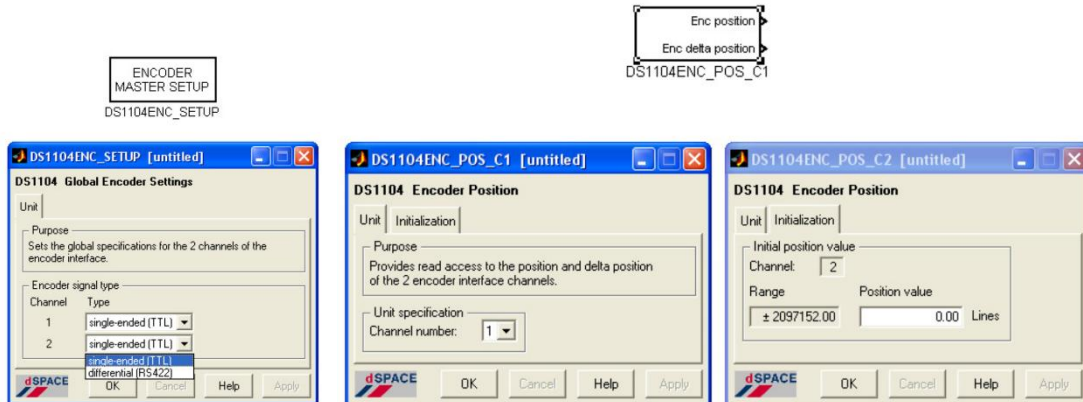


Figure 7.a

Figure 7.b

- **INDEX** : DS1104ENC\_HW\_INDEX\_C1/2 [2], [11]

Unit : Channel number 1 or 2

Parameters :

Action on index detection (Reset position count) : never / once / always

Output after index detection : kipt high / current index

Choisir 'always and current index'.

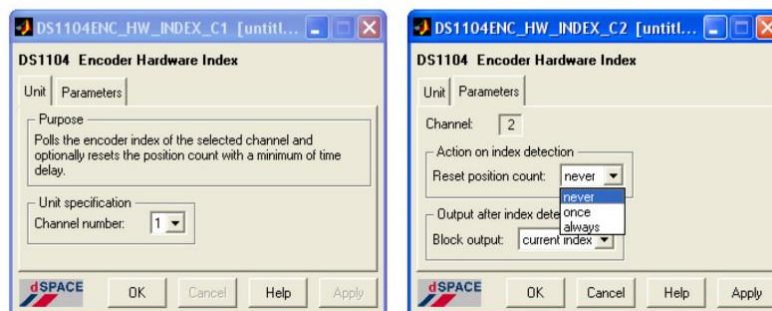
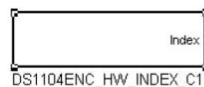


Figure 8

## c. Adaptation des unités des mesures [2], [11]

Le nombre d'impulsions (points) par tour de l'encodeur est  $N = 1024$  points ( $\approx 1000$ ), d'où :

$$n_{encod} = 0 \rightarrow \theta = 0 \text{ rad} \quad \text{et} \quad n_{encod} = 1000 \rightarrow \theta = 2\pi \text{ rad}$$

où  $n_{encod}$  est la valeur du compteur incrémental activé par les signaux encodeurs.

La carte permet la détection des fronts montants et descendants des signaux A et B.

D'où les résultats suivants [2] :

- Résolution :  $4 \times N = 4000$
- Précision :  $\frac{2\pi}{N} \cdot \frac{1}{4} = \frac{2\pi}{4000} = 0,00157 \text{ rad}$  ou  $\frac{360^\circ}{4000} = 0,09^\circ$

Pas du compteur incrémental : 0,25 point.

Pour avoir une mesure de la position angulaire en rad il faut utiliser le gain :  $ka = \frac{2\pi}{1000}$

La mesure de vitesse est obtenue en utilisant le gain :

$$ks = \frac{2\pi}{1000} \times \frac{1}{T_e} \text{ pour l'unité rad/mn} \quad \text{ou} \quad ks = \frac{60}{1000} \times \frac{1}{T_e} \text{ pour l'unité tr/mn}$$

$T_e$  est la période d'échantillonnage (la valeur choisie dans ce travail est  $T_e = 10^{-4} \text{ s}$ ).

Afin d'alléger la procédure de calcul, les deux gains sont introduits en valeurs numériques.

### • Construction du modèle Simulink

Le modèle temps réel a implanté dans Simulink est celui de la figure 9 ci-dessous :

- Démarrer le programme Matlab, puis choisir la carte dSPACE DS1104 utilisée (**RTI1104**).
- Ouvrir un nouveau fichier Simulink, puis charger la **librairie RTI1104** à partir de **Simulink Library Browser** (ou en tapant la commande RTI1104).
- Créer **votre propre dossier de travail** sous un nom de la forme suivante : **TP03\_Gr\_SGr\_Initiales**.
- Créer **dans votre dossier de travail** un sous dossier sous le nom : **TP03\_MAS**.
- Planter le modèle Simulink de la figure 9, en y insérant les blocs fonctionnels nécessaires et disponible dans la bibliothèque Simulink et la librairie **Master PPC**.
- Déterminer, puis introduire dans le modèle les valeurs des gains pour chacun des capteurs de courants ( $G_{i1}$  et  $G_{i2}$ ) et de tension ( $G_v$ ).
- Déterminer, puis introduire la valeur de l'offset de chacun des capteurs de courant.
- Configurer la valeur de  $T_e$  dans **File -> Model properties -> Model properties -> Callbacks -> initFunction**.
- Sauvegarder votre fichier Simulink sous le nom **mas\_dem.slx** dans votre sous dossier.
- Compiler le programme.
- Vérifier que la procédure de génération du code C s'est bien déroulée.

### • Création de l'interface sous ControlDesk et visualisation des sorties

- Lancer le logiciel ControlDesk NG et créer une interface utilisateur (expérience et tableau de bord) permettant de visualiser en temps réel les différentes variables du système.
- Choisir les instruments nécessaires à la visualisation de la tension  $u_{bc}$ , les trois courants  $i_a$ ,  $i_b$  et  $i_c$ , la vitesse ( $\Omega$  en tr/mn et  $N$  en rad/s).

### • Mesures, acquisition et sauvegarde

- Régler l'alimentation sur 400 Volts (entre phase) avant le démarrage du moteur.
- Lancer l'exécution du programme en cliquant sur **Go Online**, puis l'acquisition des mesures en (**mode Measuring**) cliquant sur le bouton **Start Measuring**.
- Lancer la sauvegarde des variables : tension  $u_{bc}$ , courants  $i_a$ ,  $i_b$ ,  $i_c$ , vitesse  $N$  et pulsation  $\Omega$ .
- Démarrer le moteur en alimentant le montage.
- Après quelques secondes, couper l'alimentation du moteur.
- Exporter le fichier (**.mat**) (Nommer le : **mas\_01.mat**), sauvegarder le dans le dossier de travail),
- Cliquer sur **Go Offline**, puis couper l'alimentation du moteur.
- Extraire les variables (tension, courants et vitesses) en modifiant le script de récupération des données.
- Tracer les chronogrammes obtenus et interpréter les allures.

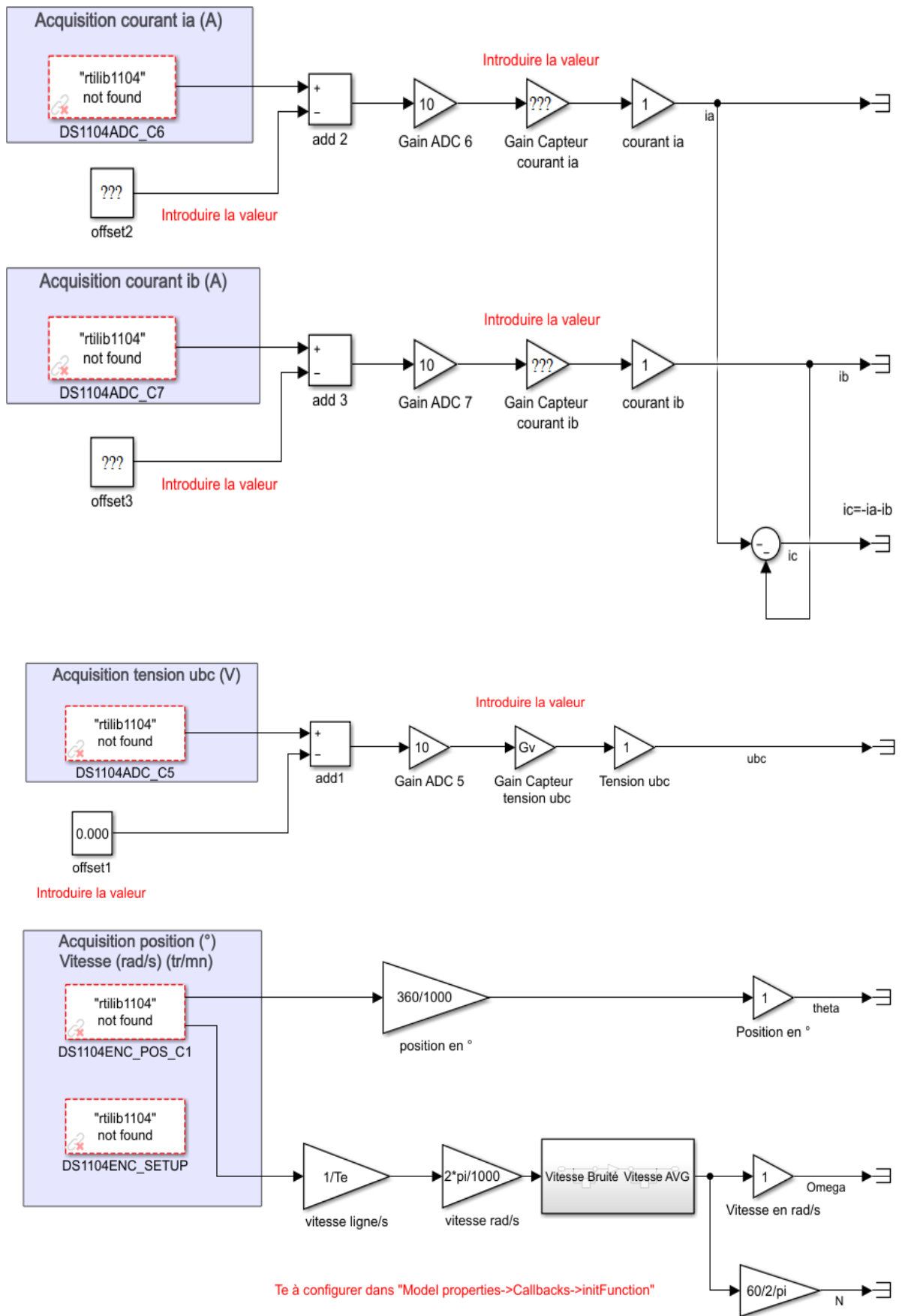


Figure 9

## 4. Démarrage direct d'un moteur à courant continu

### • Présentation du dispositif expérimental

Le schéma du dispositif expérimental étudié est représenté sur la figure 10.

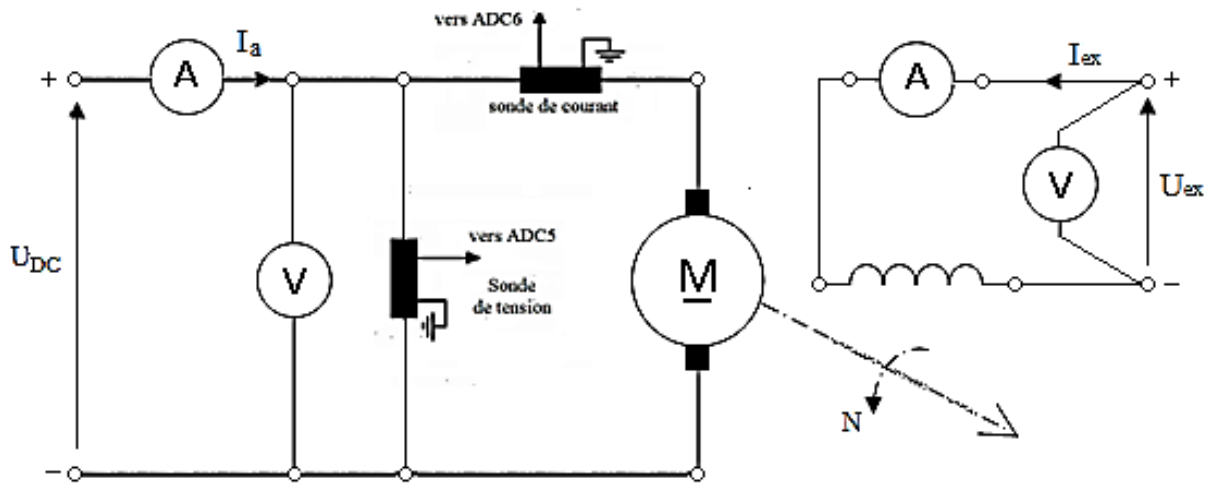


Figure 10

La fréquence d'échantillonnage  $f_e$  choisie sera de 10 kHz ( $T_e = 0,1$  ms) (Il faut que  $f_e \gg 50$  Hz).

Les sorties des différents capteurs seront connectées aux entrées suivantes :

- Tension  $U_{DC}$  : Entrée ADC, canal ADCH5 ; ( $U_{DC}$  : Tension d'induit)
- Courant  $I_a$  : Entrée ADC, canal ADCH6 ;
- Vitesse de rotation : Entrée Inc1, ( $N$  en tr/mn) et  $\Omega$  en rad/s).

$U_{ex}$  et  $I_{ex}$  sont respectivement la tension d'alimentation de l'inducteur et l'intensité du courant d'excitation.

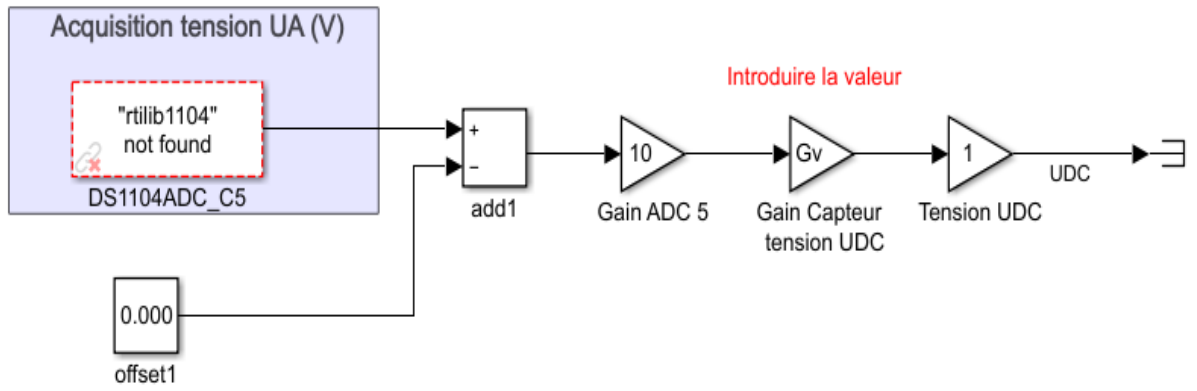
### • Construction du modèle Simulink

Le modèle temps réel a implémenté dans Simulink est celui de la figure 11 ci-dessous :

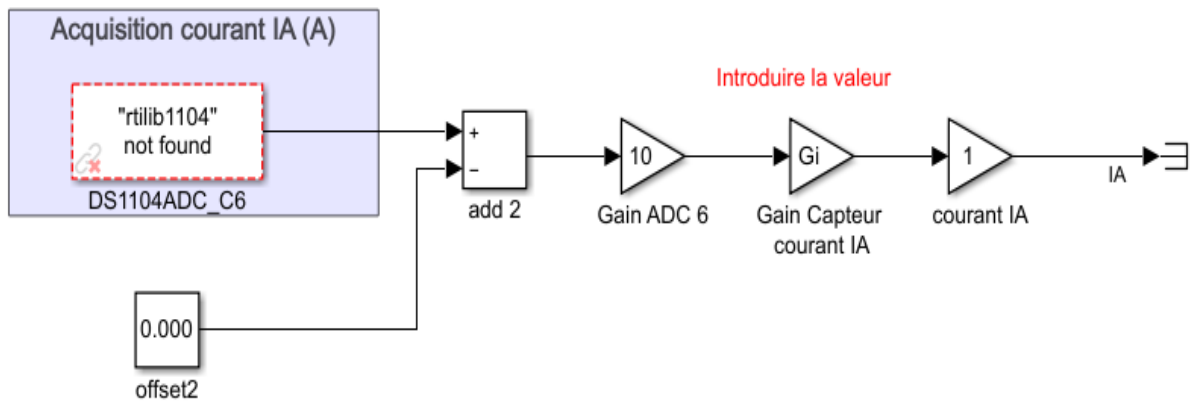
- Démarrer le programme Matlab, puis choisir la carte dSPACE DS1104 utilisée (**RTI1104**).
- Ouvrir un nouveau fichier Simulink, puis charger la **librairie RTI1104** à partir de **Simulink Library Browser** (ou en tapant la commande RTI1104).
- Créer **dans votre dossier de travail** un sous dossier sous le nom : **TP03\_MCC**.
- Implanter le modèle Simulink de la figure 11, en y insérant les blocs fonctionnels nécessaires et disponible dans la bibliothèque Simulink et la librairie Master **Master PPC**.
- Configurer la valeur de  $T_e$  dans **File -> Model properties -> Model properties -> Callbacks -> initFunction**.
- Sauvegarder votre fichier Simulink sous le nom **mcc\_dem.slx** dans votre sous dossier.
- Compiler le programme.
- Vérifier que la procédure de génération du code C s'est bien déroulée.

### • Création de l'interface sous ControlDesk et visualisation des sorties

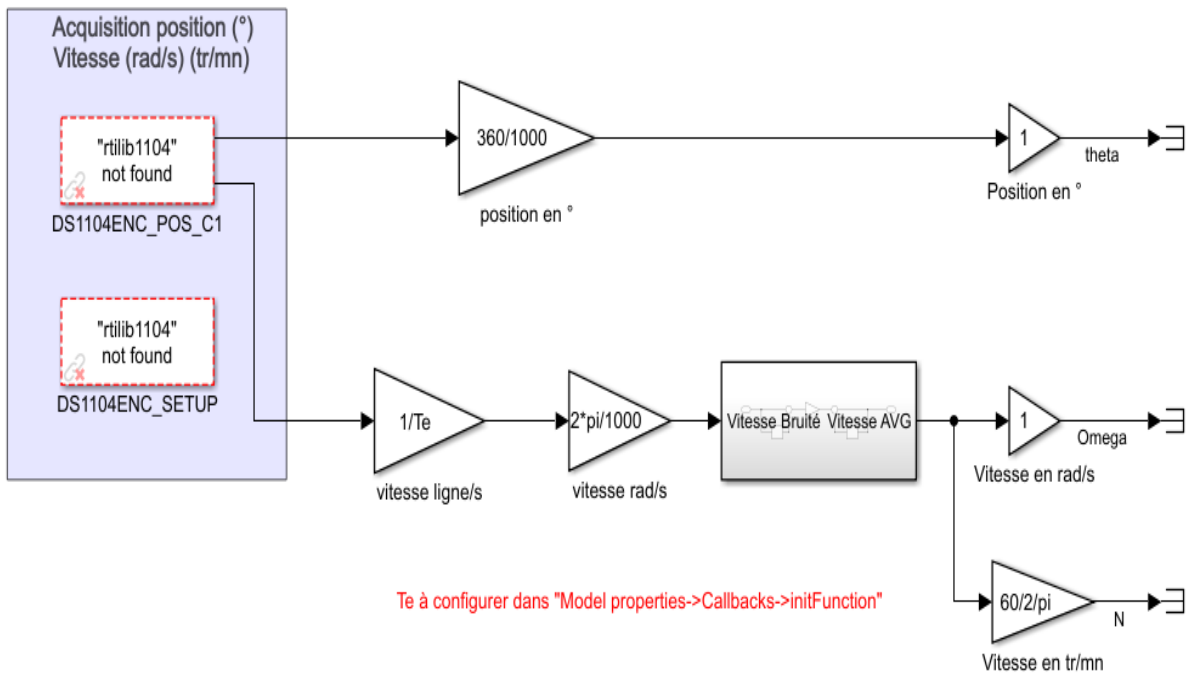
- Lancer le logiciel ControlDesk NG et créer une interface utilisateur (expérience et tableau de bord) permettant la visualisation des variables du système.
- Choisir les instruments nécessaires à la visualisation de la tension  $U_{DC}$ , le courants  $I_a$  et la vitesse ( $\Omega$  en tr/mn et  $N$  en rad/s).



Introduire la valeur



Introduire la valeur



Te à configurer dans "Model properties->Callbacks->initFunction"

Figure 11

## • Mesures, acquisition et sauvegarde

- Alimenter le circuit inducteur du moteur et ajuster le courant d'excitation à son intensité nominale.
- Régler l'alimentation d'induit sur à sa valeur nominale avant le démarrage du moteur.
- Lancer l'exécution du programme en cliquant sur **Go Online**, puis l'acquisition des mesures en (**mode Measuring**) cliquant sur le bouton **Start Measuring**.
- Lancer la sauvegarde des variables : tension  $U_{DC}$ , courant  $I_A$  et vitesse  $N$  et pulsation  $\Omega$ .
- Démarrer le moteur en alimentant le montage.
- Attendre quelques secondes, réduire la tension d'alimentation à la moitié de la tension nominale.
- puis couper l'alimentation l'alimentation du circuit d'induit du moteur.
- Après quelques secondes, cliquer sur **Go Offline**, puis couper l'alimentation du circuit d'excitation.
- Exporter le fichier (.mat) (Nommer le : **mcc\_01.mat**), sauvegarder le dans le dossier de travail),
- Cliquer sur **Go Offline**, puis couper l'alimentation du moteur.
- Extraire les variables (tension, courants et vitesses) en modifiant le programme de récupération des données.
- Tracer les chronogrammes obtenus et interpréter les allures.

## 5. Compte-rendu

Rendre votre travail, sous forme électronique, dans une archive nommée :

(**TP03\_Rapport\_Gr\_SGr\_Initiales.rar**) qui contiendra :

- le fichier (**TP03\_Nom\_Prénom\_rapport.pdf**) de votre rapport de TP comportant vos résultats (observations, interprétations, conclusions, ...etc.)
- le dossier de travail (**TP03\_Groupe\_Binome\_Initiales**) contenant les modèles Simulink temps réel, le projet ControlDesk et l'ensemble des résultats collectés le fichiers (.mat).



### Précautions d'emploi à respecter impérativement

- **A la mise sous tension :**
  - **Alimenter d'abord le circuit inducteur pour éviter l'emballement du moteur.**
- **Pour arrêter le moteur :**
  - **Cliquer sur Go Offline, puis couper l'alimentation du circuit d'induit.**
  - **Puis ramener le courant d'inducteur à son minimum en diminuant la tension d'excitation.**
  - **Enfin, couper l'alimentation du circuit d'excitation.**

***Fermer les applications ouvertes, éteindre l'ordinateur et couper l'alimentation du poste de travail.***

***Ranger le poste de travail et le remettre le dans l'état dans lequel vous l'avez trouvé en début de séance.***



## TP 04 : Implémentation des techniques MLI pour onduleur triphasé et hacheur Démarrage d'un MAS avec onduleur - Démarrage MCC avec hacheur

### 1. Objectifs

L'objectif de ce TP consiste à :

- se familiariser à l'utilisation, en fonctionnements onduleur et hacheur, du convertisseur SEMIKRON Semiteach à IGBT associé à la carte dSPACE DS1104 ;
- implanter une technique de MLI (PWM) pour commander un onduleur de tension devant alimenter un moteur asynchrone triphasé ;
- implanter une technique de MLI (PWM) pour commander un hacheur devant alimenter un moteur courant continu ;
- effectuer l'acquisition et la visualisation, à l'aide du logiciel ControlDesk, des courbes temporelles des variables : courant, tensions et vitesses lors du démarrage des deux types de moteurs.

### 2. Matériel nécessaire

- 01 alimentation triphasé variable (3 x 0 à 430 Volts AC / 5A),
- 01 machine asynchrone triphasé à cage 400/230 V-50Hz,
- 01 machine à courant continu,
- 01 convertisseur SEMIKRON Semiteach à IGBT (utilisé en onduleur triphasé ou en hacheur),
- 02 capteurs de courant (sondes Fluke i30s) de rapport 100 mV/A, pour la mesure des courants  $i_a$  et  $i_b$  circulant dans les phases A et B,
- 01 capteur de tension (sonde différentielle Metrix MX 9030) de rapport 1/20 et 1/200 pour la mesure de la tension  $u_{bc}$  entre les phases A et B du moteur,
- 01 capteur incrémental de vitesse (Encodeur) Baumer-GI355, 1024 points pour la mesure de la vitesse ou de la position de l'arbre du moteur,
- 01 voltmètre, 01 ampèremètre et 01 tachymètre (pour la vérification de la mesure de la vitesse),
- Le système dSPACE DS1104 pour le prototypage rapide et le contrôle en temps réel,
- 01 ordinateur hôte avec le logiciel Matlab/Simulink, la bibliothèque « dSPACE RT11104 » et l'environnement ControlDesk NG.

### 3. Utilisation du convertisseur SEMIKRON Semiteach

#### • Utilisation en onduleur triphasé

C'est un convertisseur à IGBT multifonctionnel, comportant principalement un redresseur triphasé avec condensateurs de filtrage et un onduleur de tension à deux niveaux avec hacheur de freinage.



Figure 1.a

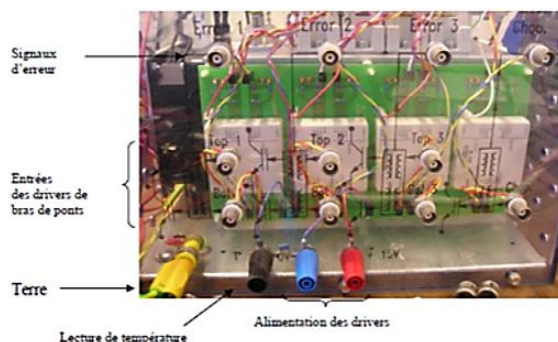


Figure 1.b

### Applications typiques :

- Éducation : peut réaliser de nombreuses applications industrielles courantes :
- Onduleur triphasé + hacheur de freinage,
- Convertisseur Buck ou boost,
- Onduleur monophasé,
- Redresseur monophasé ou triphasé.

Les interrupteurs de puissance sont équipés de drivers devant être alimentés sous 0/15V, or la carte dSPACE DS1104 délivre des signaux entre 0/5V (Niveaux TTL). L'emploi d'une interface d'adaptation convertissant la tension de 0/5V en 0/15V est donc nécessaire [19].

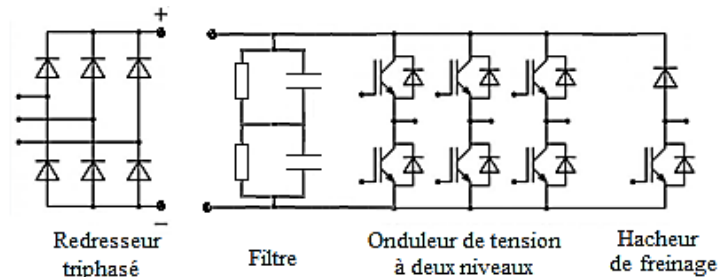


Figure 2

## 4. Démarrage d'un moteur asynchrone triphasé associé à un onduleur

### • Présentation du dispositif expérimental

Le schéma du montage expérimental est représenté sur la figure 1.

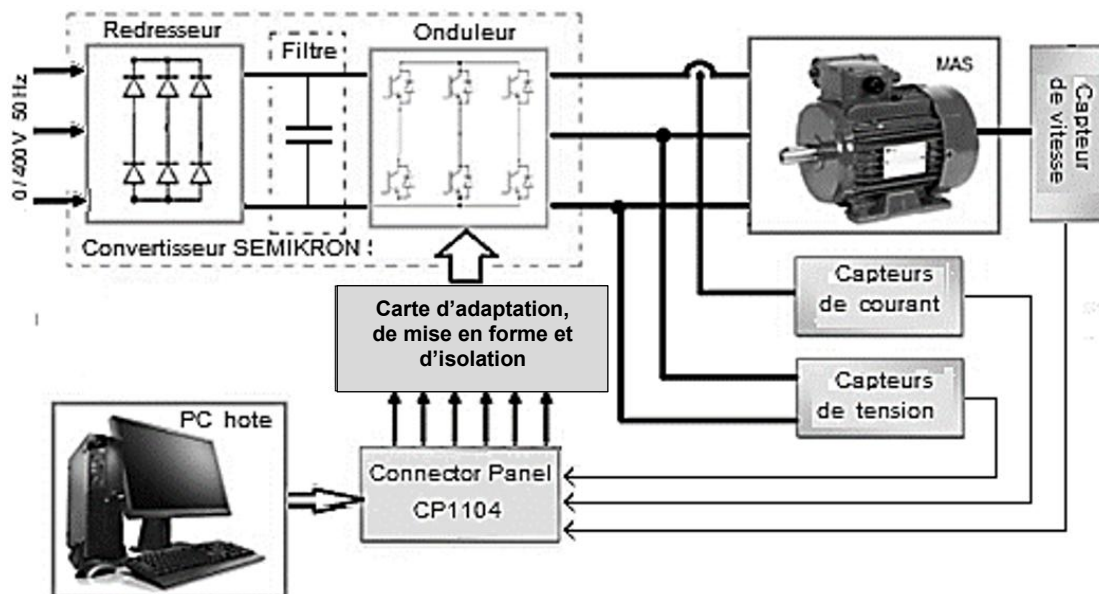


Figure 3

- Les courants  $i_a$  et  $i_b$  circulant dans les phases A et B seront mesurés, le troisième courant  $i_c$  circulant dans la phase C ne sera pas mesuré, mais déduit des deux autres courants.
- La fréquence d'échantillonnage  $f_e$  choisie sera de 10 kHz ( $T_e = 0,1$  ms) (Il faut que  $f_e \gg 50$  Hz).

Les sorties des capteurs seront connectées aux entrées suivantes :

- Tension  $u_{bc}$  : Entrée ADC, canal ADCH5 ; ( $u_{bc}$  : Tension mesurée entre les phases B et C)
- Courant  $i_a$  : Entrée ADC, canal ADCH1 ;
- Courant  $i_b$  : Entrée ADC, canal ADCH2 ;
- Vitesse de rotation : Entrée Inc1, ( $N$  en tr/mn) et  $\Omega$  en rad/s).

**Attention :** Les entrées des canaux ADCH1, ADCH2, ADCH3 et ADCH4 sont multiplexées.

## • Construction du modèle Simulink

Le modèle temps réel a implanté dans Simulink est celui de la figure 4 ci-dessous :

- Analyser le modèle et relever les équations des entrées PWM\_1, PWM\_2, PWM\_3.
- Quel est la plage réglage du bloc nommé **Fréquence**.
- Quel sont les fonctions réalisées par les blocs nommés **Set/Reset** et **Calibrage**.
- Démarrer le programme Matlab, puis choisir la carte dSPACE DS1104 utilisée (**RTI1104**).
- Ouvrir un nouveau fichier Simulink, puis charger la **librairie RTI1104** à partir de **Simulink Library Browser** (ou en tapant la commande RTI1104).
- Créer **votre propre dossier de travail** sous un nom de la forme suivante : **TP04\_Gr\_SGr\_Initiales**.
- Créer **dans votre dossier de travail** un sous dossier sous le nom : **TP04\_MAS\_MLI**.
- Planter le modèle Simulink de la figure 4, en y insérant les blocs fonctionnels nécessaires et disponible dans la bibliothèque Simulink et la librairie **Master PPC**.
- La génération de la PWM est réalisée via le bloc **DS1104SL\_DSP\_PWM** afin de produire les commandes des interrupteurs de l'onduleur.
- Insérer un bloc **DS1104SL\_DSP\_PWM**, double cliquer sur le bloc et effectuer son paramétrage.
- Déterminer, puis introduire dans le modèle les valeurs des gains pour chacun des capteurs de courants ( $G_{i1}$  et  $G_{i2}$ ) et de tension ( $G_v$ ).
- Déterminer, puis introduire la valeur de l'offset de chacun des capteurs de courant.
- Configurer la valeur de  $T_e$  dans **File -> Model properties -> Model properties -> Callbacks -> initFunction**.
- Sauvegarder votre fichier Simulink sous le nom **mas\_dem.slx** dans votre sous dossier.
- Compiler le programme.
- Vérifier que la procédure de génération du code C s'est bien déroulée.

## • Création de l'interface sous ControlDesk et visualisation des sorties

- Lancer le logiciel ControlDesk NG et créer une interface utilisateur (expérience et tableau de bord) permettant la visualisation des différentes variables du système.
- Choisir les instruments nécessaires à la visualisation de la tension  $u_{bc}$ , les trois courants  $i_a$ ,  $i_b$  et  $i_c$ , la vitesse ( $\Omega$  en tr/mn et  $N$  en rad/s).

## • Mesures, acquisition et sauvegarde

- Régler l'alimentation sur 400 Volts (entre phase) avant le démarrage du moteur.
- Lancer l'exécution du programme en cliquant sur **Go Online**, puis l'acquisition des mesures en (**mode Measuring**) cliquant sur le bouton **Start Measuring**.
- Régler la variable Fréquence sur la valeur 0,5.
- Lancer la sauvegarde des variables : tension  $u_{bc}$ , courants  $i_a$ ,  $i_b$ ,  $i_c$ , vitesse  $N$  et pulsation  $\Omega$ .
- Brancher les sondes de l'oscilloscope numérique pour visualiser les signaux de commande de l'onduleur.
- Régler la variable Fréquence sur la valeur 0,50.
- Démarrer le moteur en alimentant le montage.
- Pour les valeurs 0,25 ; 0,50 ; 0,75 ; 1,00 de la variable fréquence, relever les valeurs de la vitesse.
- Pour les valeurs 0,50 et 1,00 de la variable fréquence, visualiser et sauvegarder à l'aide de l'oscilloscope les signaux de commande de l'onduleur.
- Cliquer sur **Go Offline**, puis couper l'alimentation du moteur.

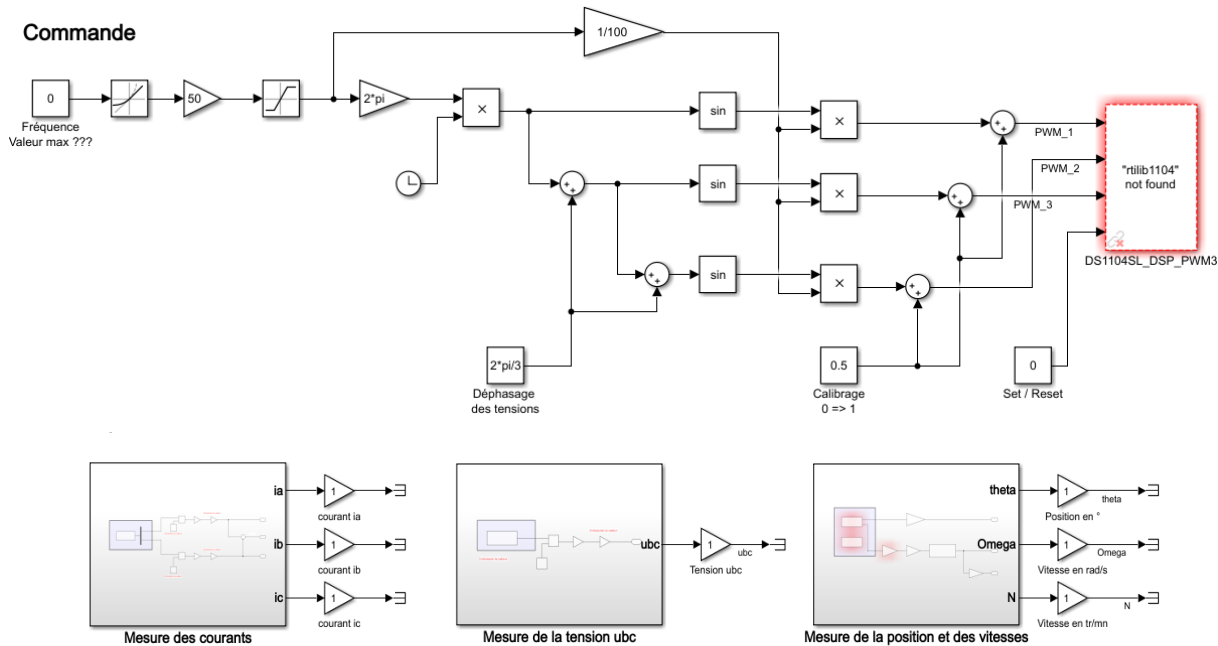


Figure 4

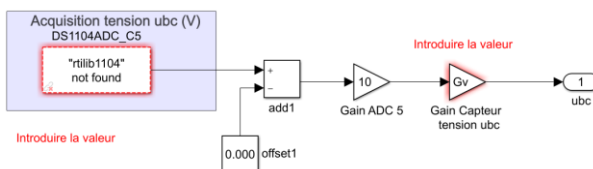


Figure 5.a

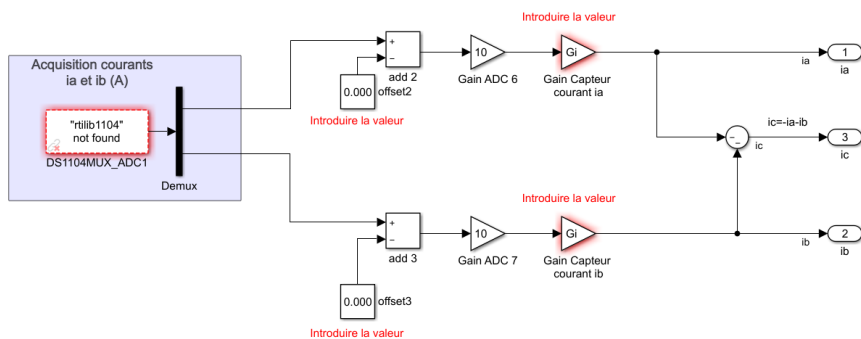


Figure 5.b

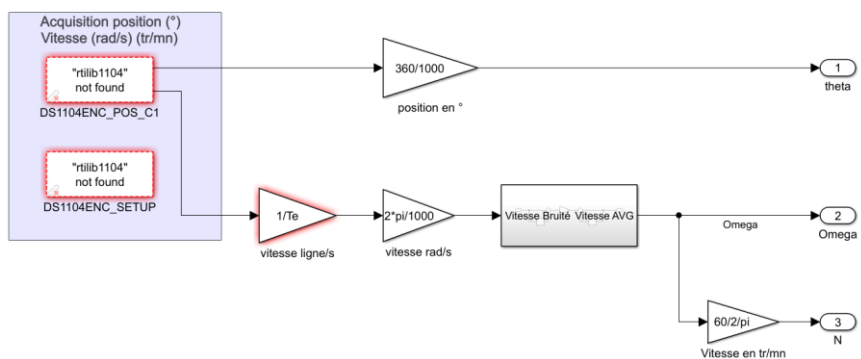


Figure 5.c

Les figures 5.a, 5.b et 5.a montrent respectivement le détail des bloc Subsystems : « **Mesure des courants** », « **Mesure de la tension ubc** » et « **Mesure de la position et des vitesses** ».

- Exporter les fichiers (.mat) (mas\_mli05.mat et mas\_mli01.mat), sauvegarder les dans le dossier de travail),
- Extraire les variables (tension, courants et vitesses) en modifiant le programme de récupération des données.
- Tracer les chronogrammes obtenus et interpréter les allures.

## 5. Démarrage d'un moteur à courant continu associé à un hacheur 4 quadrants

### • Schéma d'un hacheur en pont et fonctionnement dans les 4 quadrants

Le convertisseur de puissance de type hacheur 4 quadrants possède une configuration en pont avec deux bras. Il permet de moduler la tension d'alimentation aux bornes de la machine à courant continu et d'envisager ainsi un fonctionnement dans les différents quadrants de fonctionnement (dans les deux sens de rotation avec la possibilité de freinage).

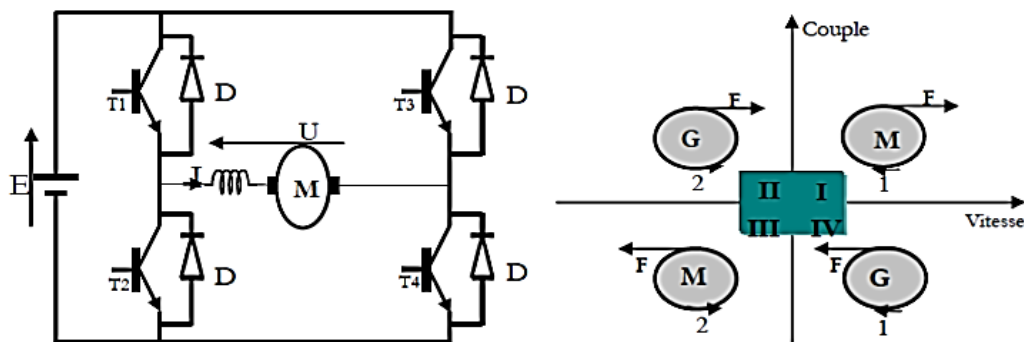


Figure 6

La figure 7 présente les allures des tension et courant :  $u(t)$  et  $i(t)$

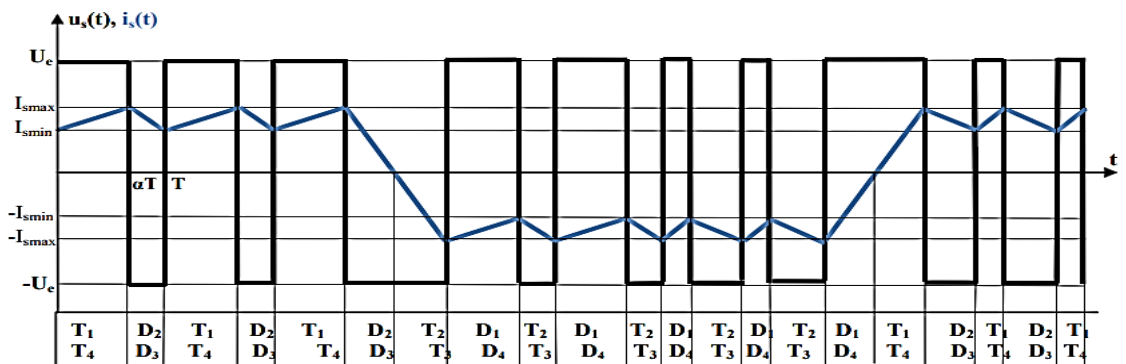


Figure 7

Pour obtenir une tension bipolaire  $u(t)$  aux bornes du moteur, les commandes des interrupteurs du même bras doivent être complémentaires et celles des interrupteurs en diagonale doivent être identiques.

- A chaque période  $T$ , on commande la fermeture des interrupteurs T1 et T4 pendant  $\alpha \cdot T$ ,
  - pendant le reste de la période, on commande la fermeture des interrupteurs T2 et T3.
- ( $\alpha$  étant le rapport cyclique)

La commande  $\bar{K}1$ , complémentaire de la commande K1, est réalisée numériquement à l'aide du bloc PWM « DS1104SL\_DSP\_PWM » de la bibliothèque RTI.

Ce convertisseur est caractérisé par une tension moyenne à sa sortie :

$$U_{moy} = (2 \cdot \alpha - 1) \cdot E$$

avec  $\alpha$  valeur du rapport cyclique et  $E$  la tension d'entrée du hacheur.

La commande des interrupteurs est réalisée selon le schéma de la figure 8.

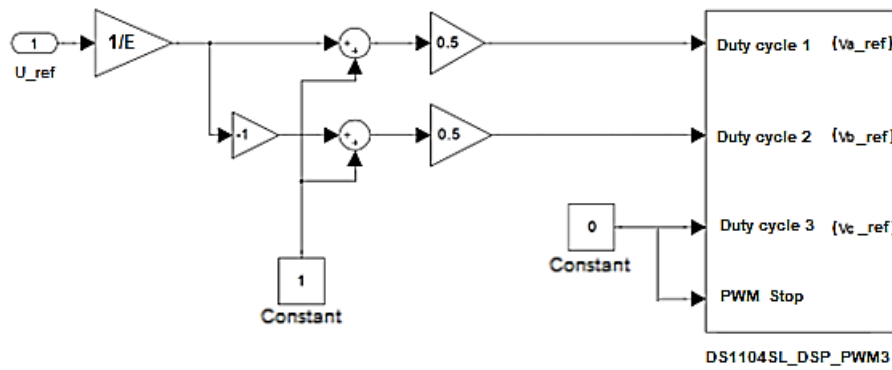


Figure 8

Les deux références  $va\_ref$  et  $vb\_ref$  ont les expressions suivantes (figure) :

$$V_{a\_PWM} = \frac{U}{2.E} + 0.5$$

$$V_{b\_PWM} = -\frac{U}{2.E} + 0.5$$

$$V_{c\_PWM} = 0$$

Le troisième bras étant inutile, on peut imposer une valeur 0 ou 1 sur la commande de  $vc\_ref$ .

### • Présentation du dispositif expérimental

Le schéma du montage expérimental est représenté sur la figure 9.

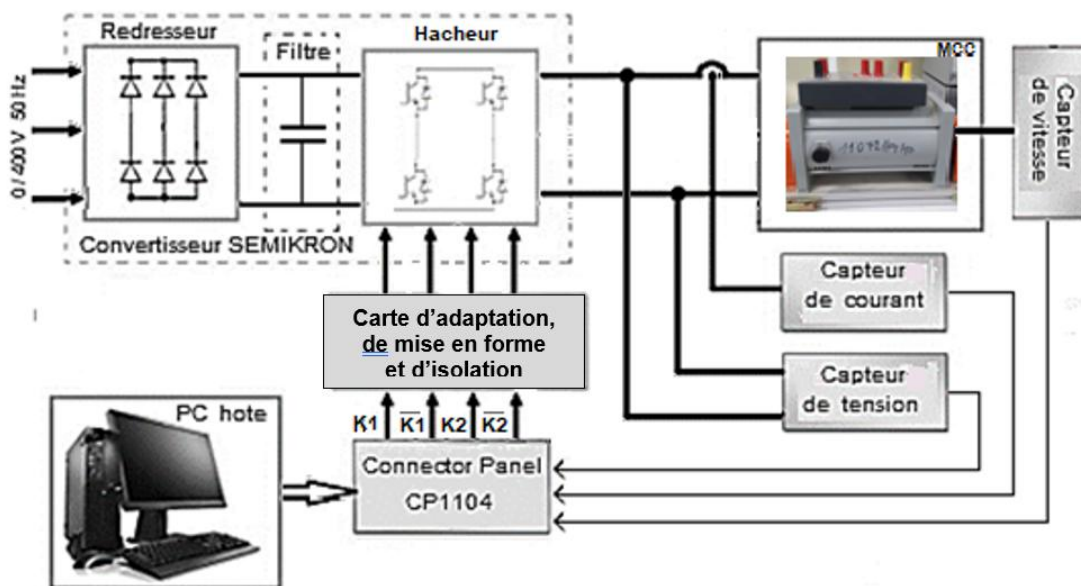


Figure 9

La fréquence d'échantillonnage  $f_e$  choisie sera de 10 kHz ( $T_e = 0,1$  ms).

Les sorties des différents capteurs seront connectées aux entrées suivantes :

- Tension  $U$  : Entrée ADC, canal ADCH5 ; ( $U_{DC}$  : Tension d'induit)
- Courant  $I_a$  : Entrée ADC, canal ADCH6 ;
- Vitesse de rotation : Entrée Inc1, ( $N$  en tr/mn) et  $\Omega$  en rad/s).

$U_{ex}$  et  $I_{ex}$  sont respectivement la tension d'alimentation de l'inducteur et l'intensité du courant d'excitation.

## • Construction du modèle Simulink

- Démarrer le programme Matlab, puis choisir la carte dSPACE DS1104 utilisée (**RTI1104**).
- Ouvrir un nouveau fichier Simulink, puis charger la **librairie RTI1104** à partir de **Simulink Library Browser** (ou en tapant la commande RTI1104).
- Créer **dans votre dossier de travail TP04\_Gr\_SGr\_Initiales**, un sous dossier sous le nom : **TP04\_MCC\_MLI**.
- Proposez un modèle Simulink temps réel destiné aux mesures de la tension  $U$  aux bornes du moteur, du courant  $I$  et de la vitesse ( $\Omega$  en tr/mn et  $N$  en rad/s).
- Faites-le vérifier puis implémenter le modèle proposé dans Matlab/Simulink, en y insérant les blocs fonctionnels nécessaires, à partir de la bibliothèque Simulink et la librairie Master **Master PPC**.
- Intégrer la génération de PWM via le bloc **DS1104SL\_DSP\_PWM** disponible dans la bibliothèque RTI afin de générer les commandes du hacheur 4 quadrants.
- Configurer la valeur de  $T_e$  dans **File -> Model properties -> Model properties -> Callbacks -> initFunction**.
- Sauvegarder votre fichier Simulink sous le nom **mcc\_dem.slx** dans votre sous dossier.
- Compiler le programme et vérifier que la procédure de génération du code C s'est bien déroulée.

## • Création de l'interface sous ControlDesk et visualisation des sorties

- Lancer le logiciel ControlDesk NG et créer une interface utilisateur (expérience et tableau de bord) permettant la visualisation des différentes variables du système.
- Choisir les instruments nécessaires à la visualisation de la tension  $U$ , aux bornes du moteur, son courant  $I$  et la vitesse ( $\Omega$  en tr/mn et  $N$  en rad/s).
- Ajouter des **sliders** sur le Layout pour pouvoir changer les rapports cycliques (duty cycle 1 et duty cycle 2).

## • Mesures, acquisition et sauvegarde

- Les commandes PWM du hacheur seront observées et sauvegardées sur l'oscilloscope numérique.
- Alimenter le circuit inducteur du moteur et régler le courant d'excitation à son intensité nominale **avant le démarrage**.
- Lancer l'exécution du programme en cliquant sur **Go Online**,
- Démarrer le moteur (Commande PWM\_Stop)
- Augmenter très doucement la tension de référence en observant le courant et la vitesse du moteur.
- **Si le moteur s'emballe, remettez très rapidement, la référence de tension à 0.**
- Régler l'alimentation en entrée du hacheur sur la valeur nominale du moteur.
- Pour trois valeurs de la tension de référence, démarrer le moteur et faites l'acquisition des mesures et la sauvegarde des variables.
- Pour chaque essai, visualiser et sauvegarder l'évolution de la tension, du courant et de la vitesse.
- Cliquer sur **Go Offline**, puis diminuer et couper l'alimentation du circuit d'induit.
- Diminuer puis couper le courant d'excitation.
- Exporter les fichiers (.mat) (**mcc\_mli1.mat**, **mcc\_mli2.mat** et **mcc\_mli2.mat**), sauvegarder les dans le dossier de travail),
- Extraire les variables (tension, courants et vitesses) en modifiant le programme de récupération des données.
- Tracer les chronogrammes obtenus et interpréter les allures.

## 6. Compte-rendu

Rendre votre travail, sous forme électronique, dans une archive nommée : **(TP04\_Rapport\_Gr\_SGr\_Initiales.rar)** qui contiendra :

- le fichier (**TP0X\_Rapport\_Gr\_SGr\_Initiales.pdf**) de votre rapport de TP comportant vos résultats (observations, interprétations, conclusions, ...etc.)
- le dossier de travail (**TP04\_Gr\_SGr\_Initiales**) contenant les modèles Simulink temps réel, le projet ControlDesk et l'ensemble des résultats collectés le fichiers (**.mat**).



### Précautions d'emploi à respecter impérativement

- **A la mise sous tension :**
  - **Alimenter d'abord le circuit inducteur pour éviter l'emballement du moteur.**
- **Pour arrêter le moteur :**
  - **Cliquer sur Go Offline, puis couper l'alimentation du circuit d'induit.**
  - **Puis ramener le courant d'inducteur à son minimum en diminuant la tension d'excitation.**
  - **Enfin, couper l'alimentation du circuit d'excitation.**

***Fermer les applications ouvertes, éteindre l'ordinateur et couper l'alimentation du poste de travail.***

***Ranger le poste de travail et le remettre le dans l'état dans lequel vous l'avez trouvé en début de séance.***



# TP 05 : Implémentation de la commande d'un moteur à courant continu

## Commande en couple et en vitesse

### 1. Objectifs

L'objectif de ce TP consiste à :

- implanter, sur la carte dSPACE DS1104, la structure de commande d'une chaîne de conversion électromécanique constituée par une source d'alimentation triphasée, un convertisseur de puissance, un moteur à courant continu et sa charge mécanique ;
- implanter une technique de MLI (PWM) pour piloter un hacheur 4 quadrants devant alimenter le moteur à courant continu ;
- mettre en place une régulation du courant d'induit pour effectuer la commande en couple basée sur un régulateur PI ;
- mettre en place une régulation de vitesse par la mise en cascade du régulateur de courant et d'un régulateur PI de vitesse ;
- effectuer l'acquisition et la visualisation, à l'aide du logiciel ControlDesk, des courbes temporelles des variables : courant, tension et vitesse, lors du démarrage du moteur sous différentes consignes de vitesse.

### 2. Matériel nécessaire

- 01 alimentation triphasée variable (3 x 0 à 430 Volts AC / 5A),
- 01 à courant continu (MCC), utilisée en fonctionnement moteur avec l'excitation séparée,
- 01 machine synchrone utilisée en mode génératrice
- 01 plan de charge résistif variable afin de produire un couple résistant ,
- 01 convertisseur SEMIKRON Semiteach à IGBT, utilisé en hacheur 4 quadrants, permettant de moduler la tension d'alimentation aux bornes de la MCC et d'envisager un fonctionnement les deux sens de rotation avec la possibilité de freinage,
- 01 capteur de courant (sondes Fluke i30s) de rapport 100 mV/A, pour la mesure du courant  $i$  absorbé par le moteur,
- 01 capteur de tension (sonde différentielle Metrix MX 9030) de rapport 1/20 et 1/200 pour la mesure de la tension d'alimentation aux bornes de l'induit du moteur,
- 01 capteur incrémental de vitesse (Encodeur) Baumer-GI355, 1024 points pour la mesure de la vitesse ou de la position de l'arbre du moteur,
- 01 machine synchrone triphasée à cage 400/230 V - 50Hz,
- 02 voltmètres et 02 ampèremètres magnétoélectriques,
- 01 tachymètre (pour la vérification de la mesure de la vitesse),
- 01 bobine lissage (facultative),
- 01 carte de prototypage dSPACE DS1104 avec panneau de contrôle CP1104
- 01 ordinateur hôte avec le logiciel Matlab/Simulink, la bibliothèque « dSPACE RTI1104 » et l'environnement ControlDesk NG.

### 3. Présentation du dispositif expérimental

Le schéma du dispositif expérimental étudié est représenté sur la figure 1.

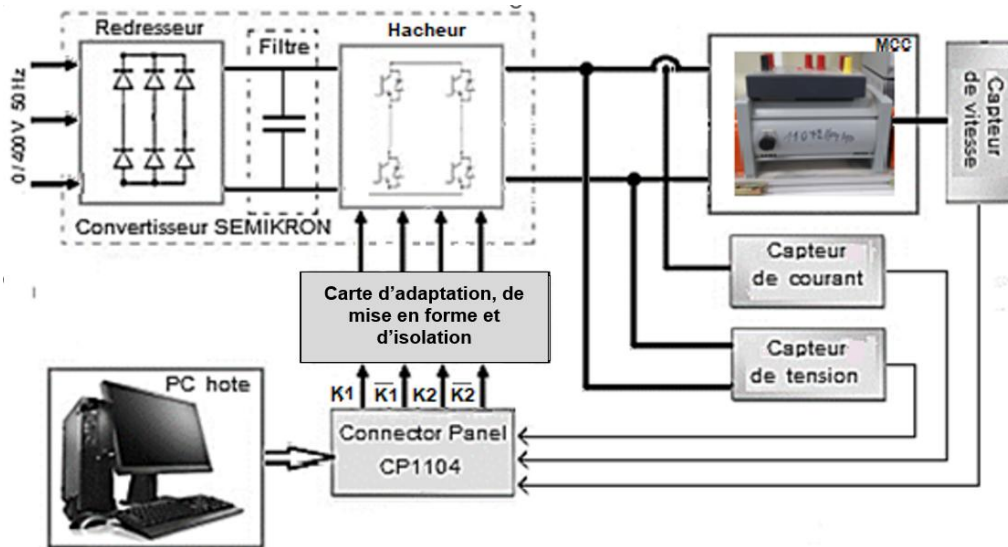


Figure 1

Les sorties des capteurs seront connectées aux entrées suivantes du panneau de connexion CP1104 :

- Tension  $U$  : Entrée ADC, canal ADCH5 ;
- Courant  $I$  : Entrée ADC, canal ADCH6 ;
- Vitesse de rotation : Entrée Inc1, ( $N$  en tr/mn) et  $\Omega$  en rad/s).

La fréquence d'échantillonnage  $f_e$  choisie sera de 10 kHz ( $T_e = 0,1$  ms).

$U_{ex}$  et  $I_{ex}$  sont respectivement la tension d'alimentation de l'inducteur et l'intensité du courant d'excitation.

Si nécessaire, on pourra rajouter une bobine lissage ( $L_b, R_b$ ) en série de la MCC pour réduire les ondulations du courant.

### 4. Commande en couple du moteur à courant continu

#### • Construction du modèle Simulink

Le modèle temps réel implanté dans Simulink est celui de la figure 2 ci-dessous :

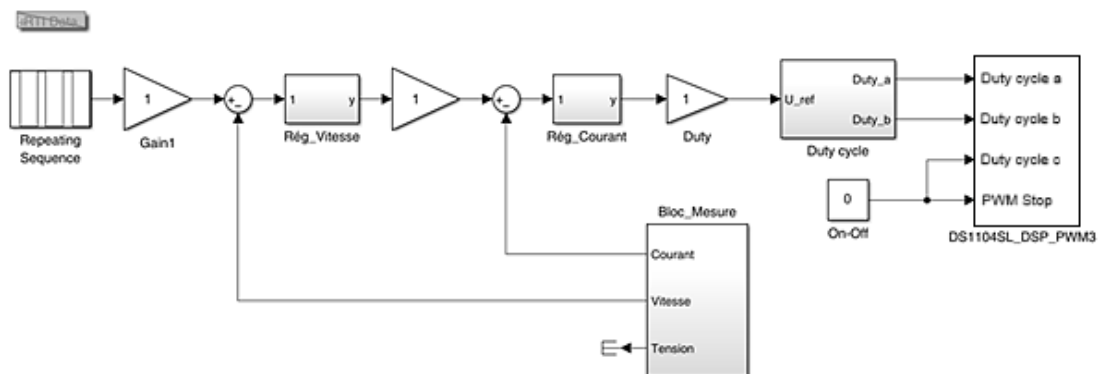


Figure 2

Le bloc **Repeating sequence** impose la référence de courant rectangulaire, le bloc **Reg\_Courant** représente le régulateur PI de courant (ou de couple), le **Bloc\_Mesure** contient le modèle permettant d'effectuer les mesures de la tension  $U$  aux bornes du moteur, son courant  $I$  et la vitesse ( $\Omega$  en rad/s et  $N$  en tr/mn).

La commande des interrupteurs est réalisée par le bloc **Duty cycle** comme le montre la figure 3.

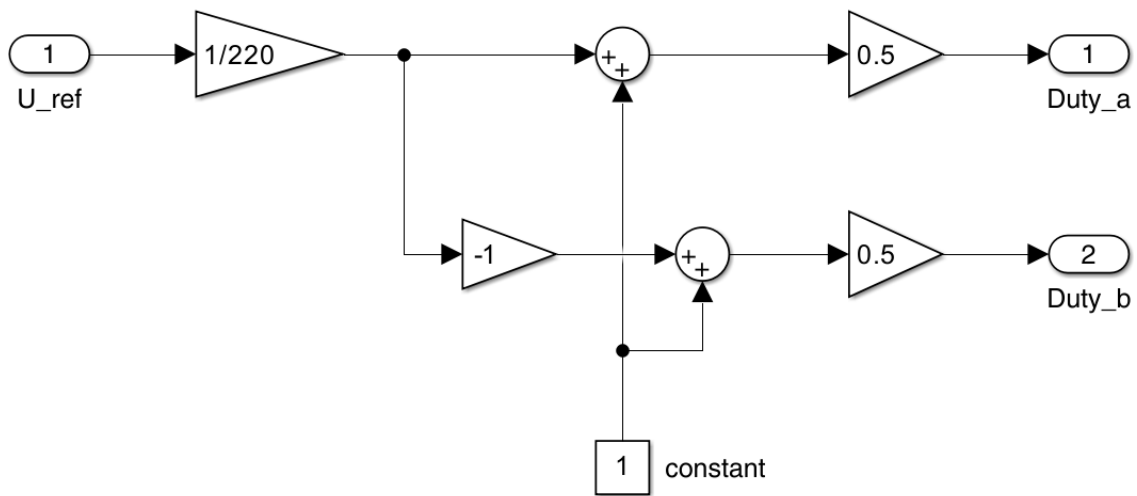


Figure 3

La génération de la PWM est faite via le bloc **DS1104SL\_DSP\_PWM3** de la bibliothèque RTI afin de générer les commandes du hacheur 4 quadrants.

- Analyser le modèle et expliquer son principe de fonctionnement.
  - Expliquer le rôle de chaque élément du modèle.
  - Démarrer le programme Matlab, puis choisir la carte dSPACE DS1104 utilisée (**RTI1104**).
  - Ouvrir un nouveau fichier Simulink, puis charger la **librairie RTI1104** à partir de **Simulink Library Browser** (ou en tapant la commande RTI1104).
  - Créer votre propre dossier de travail sous un nom de la forme suivante : **TP05\_Gr\_SGr\_Initiales**.
  - Créer **dans votre dossier de travail** un dossier sous le nom : **TP05\_MCC\_Couple**.
  - Faites du dossier **TP05\_MCC\_Couple**, votre dossier courant dans Matlab.
  - Implanter le modèle de la figure 2 dans Matlab/Simulink, en y insérant les blocs fonctionnels nécessaires et disponibles dans la bibliothèque Simulink et la librairie Master **Master PPC**.
  - Proposer un modèle pour le **bloc\_Mesure**.
  - Sauvegarder votre fichier Simulink sous le nom **mcc\_couple.slx**.
  - Configurer la valeur de  $T_e$  dans **File -> Model properties -> Model properties -> Callbacks -> initFunction**.
  - Faites vérifier le modèle implanté de la commande en couple.
  - Compiler le programme.
  - Vérifier que la procédure de génération du code C s'est bien déroulée.
- **Création de l'interface sous ControlDesk et visualisation des sorties**
    - Lancer le logiciel ControlDesk NG et créer une interface utilisateur (expérience et tableau de bord) permettant la visualisation des différentes variables du système.
    - Choisir les instruments nécessaires à la visualisation de la tension  $U$ , aux bornes du moteur, son courants  $I$  et la vitesse ( $\Omega$  en tr/mn et  $N$  en rad/s).
    - Ajouter des instruments, tel que des **sliders** et **numeric input** sur le layout pour pouvoir modifier les paramètres du bloc **Repeating sequence** et les gains du bloc **Reg\_Courant**.
    - Compléter le layout par des instruments de visualisation, tel que des **Plotter(s)**, **Gauge(s)**, ..., afin de pour visualiser certains signaux que l'on jugera pertinent.

- **Mesures, acquisition et sauvegarde**

- Paramétrer le bloc **Repeating sequence** pour imposer une référence de courant rectangulaire d'amplitude 0,5 A et de période 10 s et de rapport cyclique de 50%.
- Régler les gains du bloc **Reg\_Courant** (Régulateur PI) :  $K_p = 12$  et  $K_i = 1$ .
- Alimenter le circuit inducteur du moteur et régler le courant d'excitation à son intensité nominale avant le démarrage.
- Lancer l'exécution du programme en cliquant sur **Go Online**,
- Démarrer le moteur (Commande PWM\_Stop) et augmenter très doucement la tension l'alimentation en entrée du hacheur, en observant le courant et la vitesse du moteur.
- **Si le moteur s'emballe, remettez très rapidement, la référence de tension à 0.**
- Régler l'alimentation en entrée du hacheur sur la valeur nominale du moteur.
- Arrêter le moteur en cliquant sur **Go Offline**, puis pour l'intensité 0,4 A du courant de référence donc de couple, démarrer le moteur, faites l'acquisition et la sauvegarde des mesures de tension, de courant et de vitesse.
- Arrêter le moteur, puis redémarrer et refaites l'acquisition pour les courants de référence de 0,4 ; 0,5 A, puis de 0.6 A.
- Cliquer sur **Go Offline**, diminuer puis couper l'alimentation du circuit d'induit.
- Diminuer, puis couper le courant d'excitation.
- Sauvegarder, pour les trois cas, les fichiers (.mat) dans le dossier de travail (Nommer les : **mcc\_couple04.mat**, **mcc\_couple05.mat** et **mcc\_couple06.mat**),
- Extraire les variables (tensions, courants et vitesses) en modifiant le programme de récupération des données.
- Tracer les chronogrammes obtenus et interpréter les allures.

## 5. Commande en vitesse du moteur à courant continu

- **Construction du modèle Simulink**

Le modèle temps réel a implanté dans Simulink est celui de la figure 4 ci-dessous :

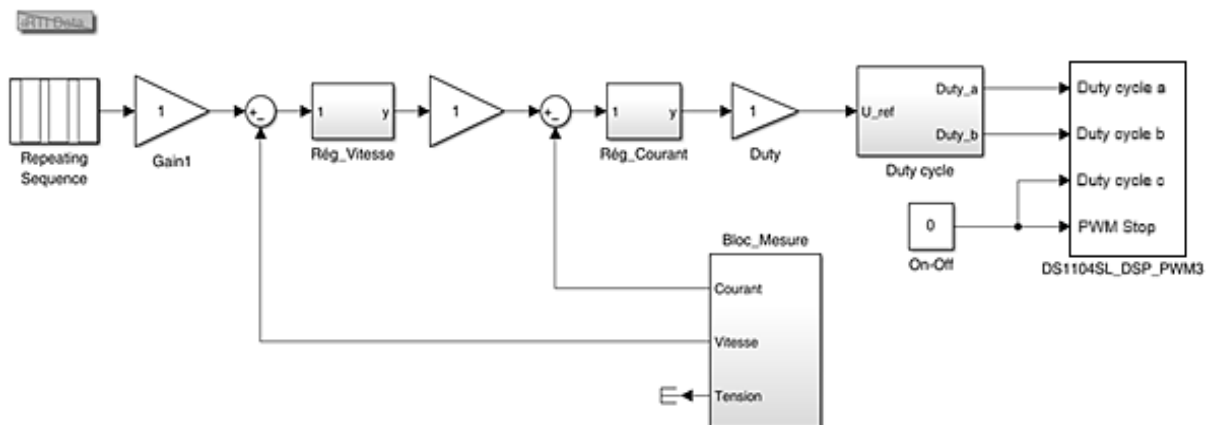


Figure 4

Le bloc **Repeating sequence** impose la référence de vitesse, le bloc **Reg\_Vitesse** représente le régulateur PI de vitesse.

- Analyser le modèle et expliquer son principe de fonctionnement.
- Expliquer le rôle de chaque élément du modèle.
- Démarrer le programme Matlab, puis choisir la carte dSPACE DS1104 utilisée (**RT11104**).

- Ouvrir un nouveau fichier Simulink, puis charger la **librairie RTI1104** à partir de **Simulink Library Browser** (ou en tapant la commande RTI1104).
- Créer **dans votre dossier de travail** un dossier sous le nom : **TP05\_MCC\_Vitesse**.
- Faites du dossier **TP05\_MCC\_Vitesse**, votre dossier courant dans Matlab.
- A partir du fichier précédent **mcc\_couple.slx**, implanter le modèle de la figure 4.
- Sauvegarder ce nouveau fichier Simulink sous le nom **mcc\_vitesse.slx**.
- Faites vérifier le modèle implanté de la commande en couple.
- Compiler le programme.
- Vérifier que la procédure de génération du code C s'est bien déroulée.

### • **Création de l'interface sous ControlDesk et visualisation des sorties**

- Lancer le logiciel ControlDesk NG et créer une interface utilisateur (expérience et tableau de bord) permettant la visualisation des différentes variables du système.
- Choisir les instruments nécessaires à la visualisation de la tension  $U$ , aux bornes du moteur, son courants  $I$  et la vitesse ( $\Omega$  en tr/mn et  $N$  en rad/s).
- Ajouter des **sliders** et **numeric input** sur le layout pour pouvoir modifier les paramètres du bloc **Repeating sequence** et les gains des bloc **Reg\_Courant** et **Reg\_Vitesse**.
- Compléter le layout par des instruments de visualisation, tel que des **Plotter(s)**, **Gauge(s)**, ..., afin de pour visualiser certains signaux que l'on jugera pertinent.

### • **Mesures, acquisition et sauvegarde**

- Paramétrer le bloc **Repeating sequence** pour imposer une référence de vitesse (Omega) rectangulaire d'amplitude 50 rad/s et de période 10 s et de rapport cyclique de 50%.
- Régler les gains du bloc **Rég\_Courant** :  $K_p = 12$  et  $K_i = 1$ .
- Régler les gains du bloc **Rég\_Vitesse** :  $K_p = 14450$  et  $K_i = 2$ .
- Alimenter le circuit inducteur du moteur et régler le courant d'excitation à son intensité nominale avant le démarrage.
- Lancer l'exécution du programme en cliquant sur **Go Online**.
- Démarrer le moteur (Commande PWM\_Stop) et augmenter très doucement la tension l'alimentation en entrée du hacheur, en observant le courant et la vitesse du moteur.
- **Si le moteur s'emballé, remettez très rapidement, la référence de tension à 0.**
- Arrêter le moteur en cliquant sur **Go Offline**, puis, faites l'acquisition et la sauvegarde des mesures de tension, de courant et de vitesse.
- Arrêter le moteur, puis redémarrer et refaite l'acquisition pour une référence de vitesse rectangulaire d'amplitude 100 rad/s, 150 rad/s et 200 rad/s.
- Cliquer sur **Go Offline**, diminuer puis couper l'alimentation du circuit d'induit.
- Sauvegarder, pour les trois cas, les fichiers (**.mat**) dans le dossier de travail (Nommer les : **mcc\_vitesse100.mat**, **mcc\_vitesse150.mat**, **mcc\_vitesse200.mat**).
- Extraire les variables (tensions, courants et vitesses) en modifiant le programme de récupération des données.
- Tracer les chronogrammes obtenus et interpréter les allures.

## 6. Compte-rendu

Rendre votre travail, sous forme électronique, dans une archive nommée :

(**TP05\_Rapport\_Nom\_Prénom.rar**) qui contiendra :

- le fichier (**TP05\_Nom\_Prénom\_rapport.pdf**) de votre rapport de TP comportant vos résultats (observations, interprétations, conclusions, ...etc.)

- le dossier de travail (**TP05\_Groupe\_Binome\_Initiales**) contenant les modèles Simulink temps réel, le projet ControlDesk et l'ensemble des résultats collectés le fichiers (.mat).



#### **Précautions d'emploi à respecter impérativement**

- **A la mise sous tension :**
  - **Alimenter d'abord le circuit inducteur pour éviter l'emballement du moteur.**
- **Pour arrêter le moteur :**
  - **Cliquer sur Go Offline, puis couper l'alimentation du circuit d'induit.**
  - **Puis ramener le courant d'inducteur à son minimum en diminuant la tension d'excitation.**
  - **Enfin, couper l'alimentation du circuit d'excitation.**

*Fermer les applications ouvertes, éteindre l'ordinateur et couper l'alimentation du poste de travail.  
Ranger le poste de travail et le remettre le dans l'état dans lequel vous l'avez trouvé en début de séance.*

# Références bibliographiques

- [1] Abdallah Darkawi, « Initiation au système dSPACE, de l'interface RTI dans Simulink et de ControlDesk Next Generation 5.4 », École d'ingénieur, France. 2016, pp.17, cel-01259564v1.  
Consultable sur le lien : [Initiation au système dSPACE, de l'interface RTI dans Simulink et de ControlDesk Next Generation 5.4](#)
- [2] Driss Yousfi, « Travaux pratiques : Alimentation et commande des moteurs DC », Département Génie Electrique, 2ème Année Cycle d'Ingénieur, Université Cadi Ayyad, Ecole Nationale des Sciences Appliquée de Marrakech.  
Consultable sur le lien : <https://fr.scribd.com/document/381190150/PolycopieTP>
- [3] M. Dahmane, O. Louhaidia, « Commandes non linéaires Hybrides d'un pendule inversé doté d'une carte Dspace », Projet de fin d'études pour l'obtention du diplôme d'ingénieur d'état en automatique, Ecole Nationale Polytechnique d'Alger, 2010.  
Consultable sur le lien : [https://repository.enp.edu.dz/jspui/bitstream/123456789/1688/1/DAHMANE.Menad\\_LOUHAIIDIA.Oussama.pdf](https://repository.enp.edu.dz/jspui/bitstream/123456789/1688/1/DAHMANE.Menad_LOUHAIIDIA.Oussama.pdf)
- [4] Salah Eddine Ababsa, Omar Mezhoudi, « Implémentation De La Commande Scalaire D'un Moteur Asynchrone Triphasé 1 kW En Utilisant La Carte Dspace 1104 », Projet de fin d'études pour l'obtention du Diplôme de Master en Electrotechnique, Université Mohamed Khider de Biskra, 2022.  
Consultable sur le lien : [Ababsa Salah Eddine-Mezhoudi Omar.pdf](#)
- [5] DS1104 R&D Controller Board Hardware Installation and Configuration For DS1104 and CP1104/CLP1104 Connector Panels, Release 4.1 - March 2004.  
Consultable sur le lien : [dSPACE DS1104 Manual | ArtisanTG](#)
- [6] Aziz Raisemche, « Commande tolérante aux défauts d'une chaine de traction d'un véhicule électrique », Automatique/Robotique, Université Paris Sud, Paris XI, 2014, NNT: 2014PA112342. tel-01127131.  
Consultable sur le lien : [2014PA112342.pdf](#)
- [7] Azad Ghaffari, « dSPACE and Real-Time Interface in Simulink », San Diego State University, Department of ECE, CA 92182-1309 12/20/2012.  
Consultable sur le lien : [https://flyingv.ucsd.edu/azad/dSPACE\\_tutorial.pdf](https://flyingv.ucsd.edu/azad/dSPACE_tutorial.pdf)
- [8] Soumia Benzarouk, Abderrazek Boulanoir, « Identification paramétrique de la machine asynchrone double alimentation », Projet de fin d'études pour l'obtention du diplôme de Master en Electrotechnique, Université Yahia Fares de Medea, 2017.  
Consultable sur le lien : <https://dSPACE.univ-medea.dz/server/api/core/bitstreams/269f458d-340c-429a-be8f-392ff112fcf0/content>
- [9] Nadia Martaj, Mohand Mokhtari, « MATLAB R2009, SIMULINK et STATEFLOW pour Ingénieurs, Chercheurs et Etudiants », Springer Heidelberg Dordrecht London New York, ISBN 978-3-642-11763-3 DOI 10.1007/978-3-642-11764-0.  
Consultable sur le lien : <https://download.e-bookshelf.de/download/0000/0146/03/L-G-0000014603-0002347083.pdf>
- [10] Nicanor Quijano, Kevin Passino, « A Tutorial Introduction to Control Systems Development and Implementation with dSPACE », Dept. of Electrical Engineering, The Ohio State University, 2015 Neil Ave, Columbus, OH 43210.  
Consultable sur le lien : [Initiation au système dSPACE, de l'interface RTI dans Simulink et de ControlDesk Next Generation 5.4 version 2.0](#)
- [11] ECE 5671/6671 – Lab 1 dSPACE DS1104 Control Workstation & Simulink Tutorial, December 17, 2018.  
Consultable sur le lien : <https://my.ece.utah.edu/~bodson/5671/Labs/Lab%201%20-%20Handout.pdf>

- [12] P.E. Kamphuis, « The dSPACE system : a development system for fast controller implementation », Master of Science Thesis, Department of Electrical Engineering, Eindhoven University of Technology, *Student thesis: Master*, 31 Oct 1996.  
Consultable sur le lien : <https://pure.tue.nl/ws/portalfiles/portal/46795106/563825-1.pdf>
- [13] Fiche technique (Semikron) : Onduleur Semiteach IGBT SKM50GB 123D+SKD51+P3/250F.  
Consultable sur le lien : [D01\\_0000.fm](https://www.semikron.com/Products/IGBTs/SKM50GB123D+SKD51+P3/250F)
- [14] Fiche technique (Chauvin Arnoux - Metrix) : Sonde de tension différentielle MX 9030.  
Consultable sur le lien : [MX9030 NF 5L](https://www.chauvinarnoux.com/fr/produit/mx9030-nf-5l)
- [15] Fiche technique (Fluke) : Pince de courant AC/DC i30s (AC/DC Current Clamp).  
Consultable sur le lien : [Pince de courant AC/DC i30s](https://www.fluke.com/fr/fr/products/energy/energy-clamp/i30s)
- [16] Datasheet : LA55-P : Current Transducer LEM LA 55-P.  
Consultable sur le lien : [LA55-P pdf, LA55-P Télécharger, LA55-P Description, LA55-P Fiches technique, LA55-P view ALLDATASHEET](https://www.lem.com/fr/fr/la55-p)
- [17] Fiche technique : Encodeur incrémental Baumer-GI355.  
Consultable sur le lien : <https://www.alldatasheet.com/datasheet-pdf/view/255342/IVO/GI355.html>

# Notes

A series of horizontal dotted lines for writing notes.

# Annexes

## A1. Carte dSPACE DS1104 : Caractéristiques techniques [1], [3], [12]

Tableau A1. Carte dSPACE DS1104 : Principales caractéristiques

Parameter		Specification
Processor		<ul style="list-style-type: none"> <li>MPC8240 processor with PPC 603e core and on-chip peripherals</li> <li>64-bit floating-point processor</li> <li>CPU clock: 250 MHz</li> <li>2 x 16 KB cache, on-chip</li> <li>On-chip PCI bridge (33 MHz)</li> </ul>
Memory	Global memory	32 MB SDRAM
	Flash memory	8 MB
Timer	4 general-purpose timers	<ul style="list-style-type: none"> <li>32-bit down counter</li> <li>Reload by hardware</li> <li>80-ns resolution</li> </ul>
	1 sampling rate timer (decrementer)	<ul style="list-style-type: none"> <li>32-bit down counter</li> <li>Reload by software</li> <li>40-ns resolution</li> </ul>
	1 time base counter	<ul style="list-style-type: none"> <li>64-bit up counter</li> <li>40-ns resolution</li> </ul>
Interrupt controller		<ul style="list-style-type: none"> <li>5 timer Interrupts</li> <li>2 incremental encoder Index line Interrupts</li> <li>1 UART interrupt</li> <li>1 slave DSP interrupt</li> <li>1 slave DSP PWM interrupt</li> <li>5 A/D converter (end of conversion) Interrupts</li> <li>1 host interrupt</li> <li>4 external Interrupts (user interrupts)</li> </ul>
A/D converter	Channels	<ul style="list-style-type: none"> <li>4 multiplexed channels equipped with one sample &amp; hold A/D converter (1x16-bit)</li> <li>4 parallel channels each equipped with one sample &amp; hold A/D converter (4x12-bit)</li> <li>Note: 5 A/D converter channels (1x16-bit and 4x12-bit) can be sampled simultaneously</li> </ul>
	Resolution	<ul style="list-style-type: none"> <li>Multiplexed channels: 16 bit</li> <li>Parallel channels: 12 bit</li> </ul>
	Input voltage range	±10 V
	Conversion time	<ul style="list-style-type: none"> <li>Multiplexed channels: 2 μs<sup>1)</sup></li> <li>Parallel channels: 800 ns<sup>1)</sup></li> </ul>
	Offset error	±5 mV
	Gain error	<ul style="list-style-type: none"> <li>Multiplexed channels: ±0.25%</li> <li>Parallel channels: ±0.5%</li> </ul>
	Offset drift	40 μV/K
	Gain drift	25 ppm/K
	Signal-to-noise ratio	<ul style="list-style-type: none"> <li>Multiplexed channels: &gt;80 dB</li> <li>Parallel channels: &gt;65 dB</li> </ul>
	D/A converter	Channels
Resolution		16-bit
Output range		±10 V
Settling time		Max. 10 μs (full-scale, accuracy ½ LSB)
Offset error		±1 mV
Gain error		±0.1%
Offset drift		130 μV/K
Gain drift		25 ppm/K
Signal-to-noise ratio		>80 dB
I <sub>max</sub>		±5 mA
Digital I/O	Channels	<ul style="list-style-type: none"> <li>20-bit parallel I/O</li> <li>Single bit selectable for input or output</li> </ul>
	Voltage range	TTL input/output levels
	I <sub>out, max</sub>	±5 mA

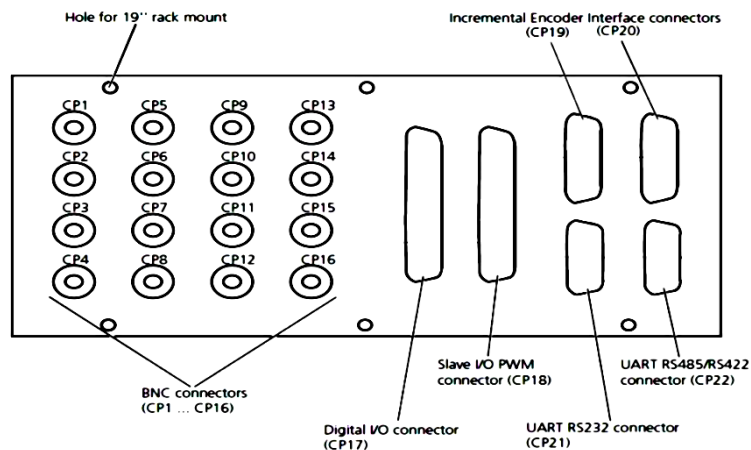
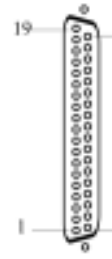


Figure A1. Description du panneau de connexion (Control Panel) CP1104

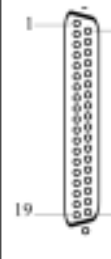
Tableau A2 : Tableaux des connexions du control panel CLP 1104

Connecteur	Signal	Connecteur	Signal	Connecteur	Signal	Connecteur	Signal
CP1	ADCH1	CP5	ADCH5	CP9	DACH1	CP13	DACH5
CP2	ADCH2	CP6	ADCH6	CP10	DACH2	CP14	DACH6
CP3	ADCH3	CP7	ADCH7	CP11	DACH3	CP15	DACH7
CP4	ADCH4	CP8	ADCH8	CP12	DACH4	CP16	DACH8

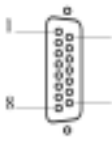
Connecteur (CP17)	Pin	Signal	Pin	Signal
	19	GND		
	18	GND	37	VCC (+5 V)
	17	GND	36	VCC (+5 V)
	16	GND	35	GND
	15	IO19	34	GND
	14	IO17	33	IO18
	13	GND	32	IO16
	12	IO15	31	GND
	11	IO13	30	IO14
	10	GND	29	IO12
	9	IO11	28	GND
	8	IO9	27	IO10
	7	GND	26	IO8
	6	IO7	25	GND
	5	IO5	24	IO6
	4	GND	23	IO4
	3	IO3	22	GND
	2	IO1	21	IO2
	1	GND	20	IO0




Connecteur (CP18)	Pin	Signal	Pin	Signal
	1	GND		
	2	SCAP1	20	GND
	3	SCAP3	21	SCAP2
	4	GND	22	SCAP4
	5	ST2PWM	23	ST1PWM
	6	GND	24	ST3PWM
	7	SPWM1	25	GND
	8	SPWM3	26	SPWM2
	9	SPWM5	27	SPWM4
	10	SPWM7	28	SPWM6
	11	SPWM9	29	SPWM8
	12	GND	30	GND
	13	GND	31	GND
	14	GND	32	GND
	15	GND	33	GND
	16	SSIMO	34	SSOMI
	17	SCLK	35	SSTE
	18	VCC (+5 V)	36	GND
	19	VCC (+5 V)	37	GND



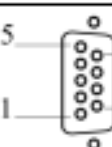
Connecteur (CP19)	Pin	Signal	Pin	Signal
	1	VCC (+5 V)		
	2	PHI0(x)	9	VCC (+5 V)
	3	PHI0(x)	10	GND
	4	PHI90(x)	11	GND
	5	PHI90(x)	12	GND
	6	IDX(x)	13	GND
	7	IDX(x)	14	GND
	8	GND	15	GND



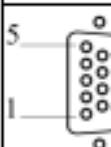
Connecteur (CP20)	Pin	Signal	Pin	Signal
	1	VCC (+5 V)		
	2	PHI0(x)	9	VCC (+5 V)
	3	PHI0(x)	10	GND
	4	PHI90(x)	11	GND
	5	PHI90(x)	12	GND
	6	IDX(x)	13	GND
	7	IDX(x)	14	GND
	8	GND	15	GND



Connecteur (CP21)	Pin	Signal	Pin	Signal
	5	GND		
	4	DTR (TXD)	9	Non utilisé
	3	TXD (TXD)	8	CTS (CTS)
	2	RXD (RXD)	7	RTS (RTS)
	1	DCD (CTS)	6	DSR (RXD)



Connecteur (CP22)	Pin	Signal	Pin	Signal
	5	GND		
	4	RXD	9	CTS
	3	RXD	8	CTS
	2	TXD	7	RTS
	1	TXD	6	RTS



## **A.2 Liste des extensions de fichier utilisées sur Control Desk [1], [3]**

**.cdx** : Projet

**.lay** : Layout

**.ppc** : Application pouvant être chargée dans le DSP

**.csv** : Fichier d'acquisition de mesures au format ASCII

**.idf** : Fichier d'acquisition de mesure en mode Stream to Disk

**.mat** : Fichier d'acquisition de mesures pouvant être chargées sur Matlab

**.sdf** : Fichier des variables qui vont être récupérées depuis Matlab

### A3. Description de l'onduleur multifonctionnel SémiKTRON [13]

- Onduleur à 2 niveaux
- Référence : Semiteach IGBT SKM50GB 123D+SKD51+P3/250F
- Double alimentation : Continue et alternative triphasée
- Cas d'une source de tensions triphasées : Redresseur incorporé avec condensateur de filtrage, pour produire une tension en continue.
- Commande de 3 bras avec 6 interrupteurs de type IGBT

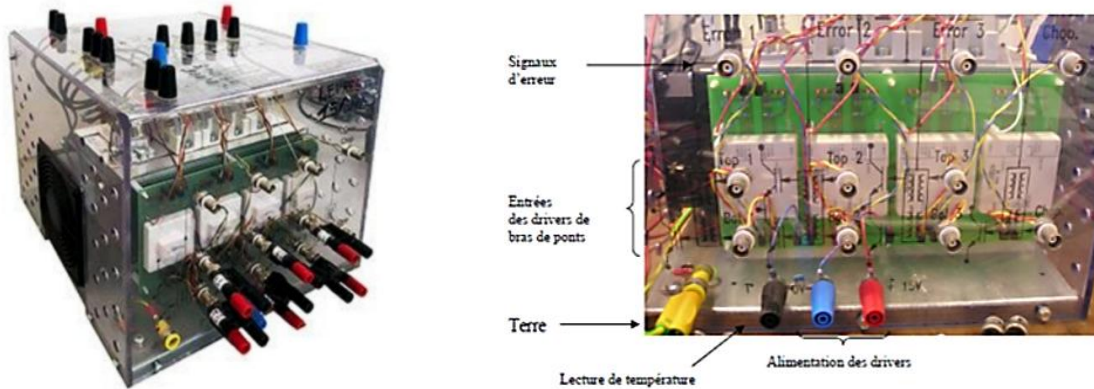


Figure A2. Vue générale du convertisseur multifonctionnel SEMIKTRON - Semiteach

#### Caractéristiques

- Convertisseur IGBT multifonctionnel.
- Boîtier transparent permettant de la visualisation de chaque pièce
- Protection IP2x pour minimiser les risques de sécurité
- Connecteurs externes de type banane/BNC pour tous les appareils Unité d'entraînement intégrée offrant détection/coupage de court-circuit, détection de défaillance de l'alimentation électrique, verrouillage des IGBTs + isolation galvanique de l'utilisateur.
- Dissipateur thermique refroidi par air forcé

#### Applications typiques

- Éducation : Peut simuler presque toutes les applications industrielles existantes :
- Onduleur triphasé + hacheur de freinage,
- Convertisseur Buck ou boost,
- Onduleur monophasé,
- Redresseur monophasé ou triphasé.

Tableau A3 : Tableaux des connexions

Symbole	Désignation	Valeur
$I_{rms}$	La valeur efficace de courant	30 A
$V_{CES}$	Tension maximale au bornes du composant	1200 V
$V_{GES}$	Tension de grille	$\pm 20$ V
$I_C$	Courant au bornes du composant	50 (40) A
$I_{CM}$	Courant maximale au bornes du composant (pour 5 $\mu$ seconds)	100 (80) A
$V_{INmax}$	Tension d'entrée maximale sans filtre/avec filtre	3x480 / 3x380 V
$C_{qvi}$	Condensateur de filtrage	1100 /800 $\mu$ F/V
$V_{DCmax}$	La valeur maximale de tension continu appliqué au condensateur	750 V
$V_c$	Tension de commande	0 / 15V
$P_I$	Consommation de courant de l'alimentation	16mA
$T_{max}$	Température maximale de fonctionnement	71°C

## A.4 Sonde de tension différentielle MX 9030 [14]

Caractéristiques techniques :

- Bande passante : 30 MHz
- Rapports de tension : 1/20 et 1/200
- Entrée : 1 voie différentielle / cordons banane amovibles
- Plage de tension différentielle d'entrée :  $\pm 0,1V$  à  $\pm 600V$
- Impédance d'entrée : 2MOHM - 6pF
- Réjection de mode commun : 80dB à 50 Hz, 50dB à 1 MHz
- Sortie Oscilloscope :  $\pm 3V$  chargée par 1 MOHM, câble BNC 1,10m environ
- Alimentation/Autonomie : 1 pile 9V type 6LF22 / 18 heures
- Sécurité/CEM : IEC 61010-1, Cat.IV | - 600V, CEM selon NF EN 61326-1
- Dimensions : 163 x 62 x 40 mm
- Poids : 195 g avec piles



Figure A3

## A5. Pince de courant AC et DC i30s [15]

Principales fonctions

- The i30s current clamp is based on Hall Effect technology for use in measurement of both DC and AC current.
- The current clamp may be used in conjunction with oscilloscopes and other suitable recording instruments for accurate non-intrusive current measurement.
- Current range: 30 mA to 30 A DC, 30 mA to 20 A AC rms
- Frequency range: DC to 100 kHz (-0.5 dB)
- CAT III 300 V safety rating



Figure A4

## A.6 Capteur de courant à effet Hall LEM LA55-P [16]

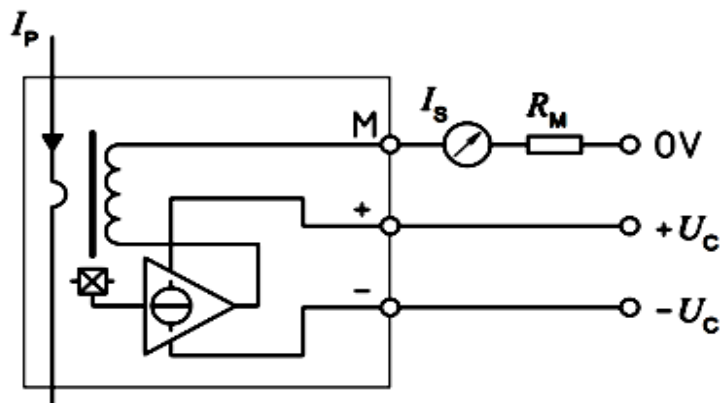


Figure A5. Capteur de courant LA55-P

- ✓ Rapport de tours 70:1
- ✓ Technologie de détection Boucle fermée
- ✓ Tension d'alimentation 12→15 V
- ✓ Courant d'entrée 70A
- ✓ Courant de sortie 50 mArms
- ✓ Hauteur hors tout 27.6mm
- ✓ Largeur hors tout 36.6mm
- ✓ Profondeur hors tout 14.45mm
- ✓ Température minimum -40°C
- ✓ Température maximum +85°C
- ✓ Série LA
- ✓ Type de montage : Montage sur CI

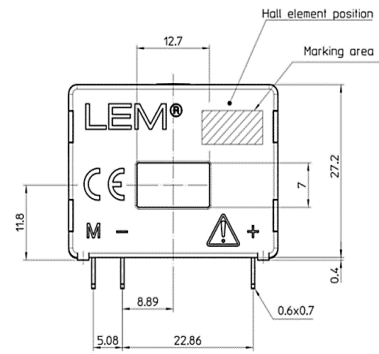


Figure A6

### Caractéristiques

- Transducteur de courant à boucle fermée (compensé) utilisant l'effet de Hall.
- Boîtier en plastique isolant reconnu norme UL 94-V0.

### Avantages

- Excellente précision
- Très bonne linéarité
- Faible dérive en température.
- Temps de réponse optimisé
- Grande largeur de bande de fréquence
- Pas de pertes d'insertion
- Haute immunité aux interférences externes
- Capacité de surcharge en courant.

### Applications

- Variateurs de vitesse et servomoteurs à courant alternatif
- Convertisseurs statiques pour les entraînements de moteurs à courant continu.
- Applications alimentées par batterie
- Alimentations sans interruption (UPS)
- Alimentations en mode commuté (SMPS)
- Alimentations pour applications de soudage applications.

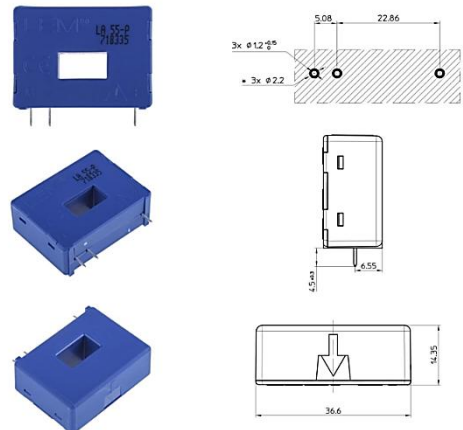


Figure A7



Figure A8. Boîtier d'un capteur de mesure pour 2 courants réalisé au LAT<sup>7</sup>

<sup>7</sup> LAT : Laboratoire d'Automatique de Tlemcen

## A.7 Encodeur incrémental Baumer-GI355 (Position / Vitesse) [17]



Figure A9. Encodeur incrémental Baumer-GI355

### Codeurs incrémentaux

Axe sortant - Bride standard ou synchro  
Résolution 5...6000 impulsions

GI355, GI356



GI355 avec bride standard

#### Caractéristiques électriques

Alimentation	5 VDC $\pm$ 10 % 4,75...30 VDC 10...30 VDC
Protection contre l'inversion de polarité	Oui (4,75...30 VDC)
Consommation à vide	$\leq$ 30 mA (24 VDC) $\leq$ 60 mA (5 VDC)
Résolution (Imp/tour)	5...6000
Signal de référence	Top zéro, largeur 90°
Principe de détection	Optique
Fréq. de commutation	$\leq$ 150 kHz
Signaux de sortie	A 90° B, 0 + compléments
Etage de sortie	Emetteur de ligne/RS422 Totem pôle, NPN et PNP
Immunité	DIN EN 61000-6-2
Emission	DIN EN 61000-6-4
Conformité	Certification UL/E63076

#### Points forts

- Codeur axe sortant  $\varnothing$ 10 mm ou  $\varnothing$ 6 mm
- Résolution max. 6000 impulsions/tour
- Détection optique
- Bride standard ou synchro
- Au standard Industriel
- Vitesse de rotation jusqu'à 10000 t/mn

#### Option

- Codeur GE355 en version INOX
- Codeur avec certification ATEX, voir modèle X 700
- Protection contre corrosion pour application offshore

#### Caractéristiques mécaniques

Dimensions (bride)	$\varnothing$ 58 mm
Charge	$\leq$ 20 N axial $\leq$ 40 N radial
Indice de protection DIN EN 60529	IP 54 (sans joint), IP 65 (avec joint)
Vitesse de rotation	$\leq$ 10000 t/min
Couple de démarrage	$\leq$ 0,015 Nm (IP 54) $\leq$ 0,03 Nm (IP 65)
Moment d'inertie rotor	14,5 gcm <sup>2</sup>
Matières	Bottier: aluminium Bride: aluminium
Température d'utilisation	-25...+100 °C (5 VDC) -25...+85 °C (24 VDC) -40...+85 °C (Option)
Humidité relative	95 % sans condensation
Résistance	DIN EN 60068-2-6 Vibration 10 g, 16-2000 Hz DIN EN 60068-2-27 Choc 200 g, 6 ms
Raccordement	Embase mâle M23, 12 points Câble 1 m
Poids	250 g
<b>GI355</b>	
Type d'axe	$\varnothing$ 10 mm axe
Bride	Bride standard
<b>GI356</b>	
Type d'axe	$\varnothing$ 6 mm axe
Bride	Bride synchro

Figure A10. Caractéristiques de l'encodeur incrémental Baumer-GI355

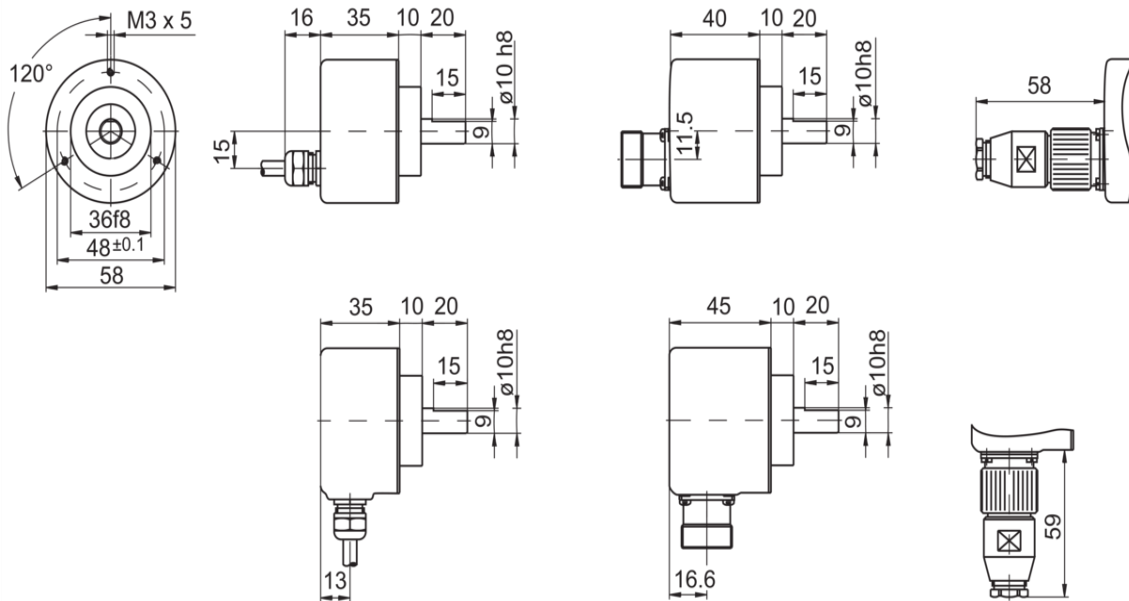


Figure A11. Encodeur incrémental Baumer-GI355

## A.8 Moteurs électriques employés

- Moteur à courant continu (Mode d'excitation : Séparée)

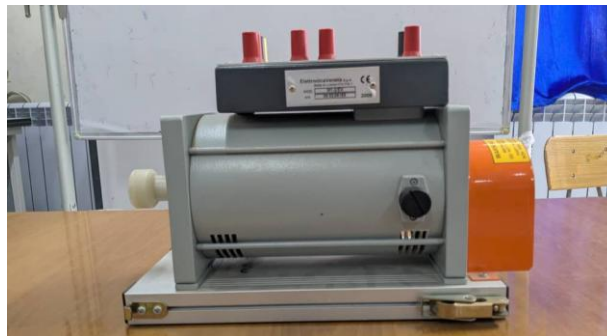


Figure A12. Vue du moteur à excitation séparée utilisé

### Caractéristiques nominales du moteur à courant continu

Tension d'induit	$U_{an} = 220 \text{ V}$	Puissance	$P_n = 380 \text{ W}$
Courant d'induit	$I_{an} = 2,5 \text{ A}$	Vitesse de rotation	$N_n = 3000 \text{ tr/mn}$
Tension d'inducteur	$U_{exn} = 220 \text{ V}$		
Courant d'inducteur	$I_{exn} = 0,1 \text{ A}$		

### Paramètres du moteur à courant continu

Résistance d'induit	$R_a = 10,51 \Omega$	Constante de couple	$K_C = 0,6902 \text{ Nm/A}$
Inductance d'induit	$L_a = 222,90 \text{ mH}$	Constante de vitesse	$K_V = 0,6902 \text{ V.s/rad}$
Résistance d'inducteur	$R_{ex} = 1945,50 \Omega$	Moment d'inertie moteur	$J_m = 0.0058 \text{ kg.m}^2$
Inductance d'inducteur	$L_{ex} = 127,67 \text{ mH}$	Coefficient de frottement	$f_v = 0,0007 \text{ Nm.s/rad}$

- **Moteur asynchrone**



Figure A13. Vue du moteur asynchrone triphasé utilisé

### Caractéristiques nominales du moteur à asynchrone

Type	Moteur à cage	Fréquence	$f = 50 \text{ Hz}$
Tensions (Y/D)	$U = 400 / 230 \text{ V}$	Puissance	$P_n = 500 \text{ W}$
Courants (Y/D)	$I = 1,50 / 2,60 \text{ A}$	Nombre de pôles	$N_p = 2$
Vitesse de rotation	$N_n = 2850 \text{ tr/mn}$	Service	S1



Figure A14. Vue d'un banc d'essai sur les quatre disponibles dans la salle de TP

