## Maquette MCC

### Sciences de l'Ingénieur

#### **DOSSIER RESSOURCE**



PRISE EN MAIN DE LA MAQUETTE	2
ESSAI EN BOUCLE OUVERTE	3
ESSAI EN BOUCLE FERMEE	4
TRACES DE COURBES AVEC PYTHON	6
MODELISATION D'UN MCC	7
SIMULATION AVEC MATLAB SIMULINK	9

#### *Conventions dans ce document*

- Indique une action à faire avec la souris ;
- Indique qu'une entrée au clavier est attendue ;
- *Indique qu'une action doit être exécutée sur le système.*

# PRISE EN MAIN DE LA MAQUETTE

En cas de dysfonctionnement, mettre l'interrupteur de mise sous tension sur 0 et relancer le logiciel de pilotage.

#### **Procédure d'initialisation :**

Ne pas mettre sous tension la maquette ;

**B**Ouvrir une session sur l'ordinateur proche de la maquette ;

Insérer l'adaptateur USB-RJ45 dans un port USB en façade de l'ordinateur ;

# Attendre la configuration de l'adaptateur par Win7 : cela peut être long !

 Insérer le câble RJ45 noir dans l'adaptateur ;
 Insérer l'autre côté du câble RJ45 dans la carte Pcduino-Nano ;

Brancher l'alimentation 9V sur la maquette ;

Placer l'interrupteur de mise sous tension sur 1 ;

#### Lancement des logiciels de pilotage :

Le moteur peut-être commandé par 2 logiciels :

#### Interface de commande en tension :

Logiciel permettant d'imposer une tension en V constante au moteur. On utilisera ce logiciel pour les essais du moteur seul. Cela correspond à des essais en boucle ouverte.

#### Interface de commande de l'asservissement en vitesse :

Logiciel permettant de piloter l'asservissement en vitesse de moteur. On utilisera ce logiciel pour les essais en boucle fermée.

Ces 2 logiciels se situent dans le menu Démarrer des PCs du laboratoire. Ils se lancent directement en double-cliquant sur les exécutables.



# **ESSAI EN BOUCLE OUVERTE**

## Réalisation de l'essai

Procéder à l'initialisation de la maquette (voir page précédente) ;
 Lancer le logiciel de pilotage de pilotage <u>Interface de commande en tension du moteur</u> ;



Régler, en validant par entrée, la consigne de tension appliquée au moteur en réglant les 3 paramètres ;

- Offset ;
- Amplitude ;
- Fréquence ;

Régler la consigne sur un signal carré pour avoir une tension en échelon ;

# **ESSAI EN BOUCLE FERMEE**

## Réalisation de l'essai

Procéder à l'initialisation de la maquette (voir page précédente) ;
 Lancer le logiciel de pilotage de pilotage <u>Interface de commande d'asservissement en vitesse</u>;



Attendre la connexion du PC avec la carte PCDuino par le port USB... cela peut être long... Attendre le défilement de valeurs dans le cadre blanc.

Régler la consigne de vitesse en réglant les 3 paramètres ;

- Offset ;
- Amplitude ;
- Fréquence ;

Sciences de l'Ingénieur

## **DOSSIER RESSOURCE**

Régler la consigne sur un signal carré pour avoir de vitesse en échelon ;

Dans la partie Asservissement, vous pouvez régler les paramètres du correcteur PID placé dans l'asservissement. Le correcteur utilisé est un PID parallèle sous la forme suivante :



Vous pouvez régler les coefficients des 3 paramètres Kp, Ki et Kd avec les 3 curseurs ;

Vous pouvez initialiser les valeurs pour revenir au réglage par défaut en cliquant sur les boutons « Ré-initialisation » ou « Ré-initialisation globale ».

# **TRACES DE COURBES AVEC PYTHON**

Le module **pyplot** de la bibliothèque **matplotlib** permet de tracer rapidement des courbes. Le principe est de placer les valeurs des abscisses et des ordonnées dans 2 listes de même longueur.

Le fichier « *ecart\_reel\_simule\_temporelle.py* », à compléter, permet de superposer deux tracés dans une même figure.

Les deux premières lignes permettent l'importation des deux bibliothèques numpy et matplotlib :

import numpy as np import matplotlib.pyplot as plt

Il faut créer la liste commune des abscisses et les listes des données des grandeurs que l'on souhaite placer en ordonnée :

t = np.array([]) # liste des valeurs du temps en secondes y1 = np.array([]) # liste des valeurs de y1 y2 = np.array([]) # liste des valeurs de y2

Pour superposer les tracés des données points par points, on utilise les commandes suivantes :

plt.plot(t,y1,'g-') # tracé de la courbe y1 en vert plt.plot(t,y2,'r-') # tracé de la courbe y2 en rouge plt.show() # montre la figure des tracés

On peut légender le graphe à l'aide des commandes suivantes :

```
plt.title('Titre du graphique') # titre du graphique
plt.xlabel('en abscisse') # titre de l'axe des abscisses
plt.ylabel('en ordonnée') # titre de l'axe des ordonnées
plt.grid(True) # mise en place d'une grille
```

Si l'on souhaite créer plusieurs figures de tracé, on peut utiliser les commandes suivantes :

<pre>fig1 = plt.figure()</pre>	#	création d'une figure de tracé
<pre>fig11 = fig1.add_subplot(1,2,1)</pre>	#	permet de créer une l <sup>ere</sup> zone de tracé dans
		un graphe
fig11.plot(x,y1)		
<pre>fig12 = fig1.add_subplot(1 ,2 ,2)</pre>	#	on créé une seconde zone
fig12.plot(x,y2)		
plt.show()		

#### Sciences de l'Ingénieur

## **Maquette MCC**

## DOSSIER RESSOURCE

## **MODELISATION D'UN MCC**



La rotation du rotor amène l'alternance du courant parcourant l'induit et donc l'alternance des pôles nord et sud du rotor, toujours en mouvement par rapport aux pôles fixes nord et sud du stator. Évidemment, il vaut mieux ne pas avoir qu'un bobinage au rotor ...

L'objectif de cette étude est de déterminer la fonction de transfert d'un moteur électrique à courant continu, reliant la vitesse de rotation entre le rotor et le stator d'une part et la tension du courant électrique fourni d'autre part. Les comportements physiques de ce type de moteur peuvent être représentés sur les deux modèles couplés suivant :



Sciences de l'Ingénieur

# **Maquette MCC**

## **DOSSIER RESSOURCE**

Les équations de couplage électromécanique permettent de relier les deux modèles ci-dessus :

$$C_{m}(t) = k.i(t)$$
  
 $e(t) = k.\omega_{m}(t)$ 

Les grandeurs caractéristiques de ces modèles sont :

- u(t) tension aux bornes de l'induit (*bobinage*) du moteur (en Volt),
- R résistance de l'induit (en Ohm),
- L inductance de l'induit (en Henry),
- k constante électromécanique du moteur (en N.m.A<sup>-1</sup> ou V.s),
- i(t) courant dans l'induit (en Ampère),
- e(t) tension contre-électromotrice,
- $\omega_m(t)$  vitesse de rotation du rotor par rapport au stator (en rad/s),
- $C_m(t)$  couple disponible sur l'arbre de sortie du moteur (en N.m),
- $C_r(t)$  couple résultant des actions mécaniques extérieures (en N.m),
- J<sub>e</sub> moment d'inertie équivalent ramené sur l'arbre de sortie du moteur (en kg.m<sup>2</sup>),
- f coefficient de frottements visqueux ramenés sur l'arbre moteur (en N.m.s<sup>-1</sup>).

# Pour la suite, on suppose le couple résistant nul : $C_r(t)=0$ .

Simulink

Library

# DOSSIER RESSOURCE

# SIMULATION AVEC MATLAB SIMULINK

#### Lancement de Simulink :

relier.

 Lancer le logiciel MATLAB-SIMULINK en double-cliquant sur l'icône MATLAB du bureau.
 Une fois que MATLAB est ouvert, cliquer sur l'icône « Simulink Library » dans la barre de navigation.

## Création d'un modèle sous la forme de schéma-bloc :

3. Dans la fenêtre « Simulink Library Browser », créer un nouveau modèle en cliquant sur « New Model ».

 **B** Simulink Library Browser

Enter search term AL -. Pour vos modèles, vous utiliserez une **Bibliotheque de SI** bibliothèque de blocs prédéfinie appelée « Bibliothèque de SI » disponible dans la Simulink Bibliotheque de SI fenêtre « Simulink Library Browser ». >In1 Out1 Communications System Toolbo Computer Vision System Toolbo Subsystem Control System Toolbox Cette bibliothèque contient les blocs les plus DSP System Toolbox courants que vous allez rencontrer dans les HDL Coder ) du/dt ) Image Acquisition Toolbox systèmes. Instrument Control Toolbox Derivative Neural Network Toolbox Simscape 4. Glisser-déplacer dans la fenêtre du modèle Simulink 3D Animation les blocs que souhaitez utiliser pour votre Simulink Coder Simulink Control Design modèle. Gain Simulink Extras Simulink Support Package for Ar ► Simulink Support Package for LI 5. Placer les blocs dans la fenêtre du modèle Stateflow System Identification Toolbox et relier les directement en cliquant sur la Integrator **Recently Used Blocks** sortie d'un bloc puis l'entrée du second bloc à

6. Affecter les bonnes valeurs numériques en double-cliquant dans chacun des blocs et en modifiant les paramètres.



Plusieurs remarques :

- le bloc Transfert **Fcn** permet définir une fonction de transfert sous la forme d'une fraction rationnelle ;
- le bloc Scope permet de définir une sortie et de l'afficher dans un graphe ;
- le paramètre de Laplace est noté s au lieu de p.

#### Configurer et lancer une simulation :

Si votre schéma-bloc est bien construit, les entrées, les sorties et tout les blocs étant définis, vous pouvez passer à la simulation de votre qui calculera numériquement toutes les valeurs à afficher dans les Scopes.

7. **Ouvrir la configuration** de la simulation temporelle en allant dans la barre transversale du modèle et en ouvrant le Menu « Simulation » puis « Model Configuration Parameters ».

8. Modifier la durée de la simulation dans la fenêtre « Simulation Time ».

9. Pour **modifier le pas de calcul**, choisir l'option « Fixed-Step » de la fenêtre « Solver options » puis indiquer le pas de calcul dans le champ « Fixed-Step Size (fundamental sample time). Valider par OK.

10. Lancer la simulation en cliquant sur l'icône « Run » de la barre transversale du modèle.



11. **Double-cliquer** dans le scope dont vous voulez visualiser le graphe.

#### Récupérer les valeurs dans un fichier :

12. **Rajouter le bloc « To Workspace »** dans votre schéma et relier la sortie du système à l'entrée de ce bloc.

13. **Paramétrer** le bloc « To Workspace » en choisissant « Structure with time » dans le champ « Save format ».

14. **Relancer** la simulation en cliquant sur « Run ».

15. **Basculer** sur la fenêtre MATLAB et **double-cliquer** sur la variable « simout » de la fenêtre « Workspace ».

Cette variable contient 2 variables : « time » et « signals ».

16. **Copier les deux colonnes** de valeurs des variables « simout.time » et « simout.signals.values » dans un fichier texte.

17. Sauvegarder le fichier texte.