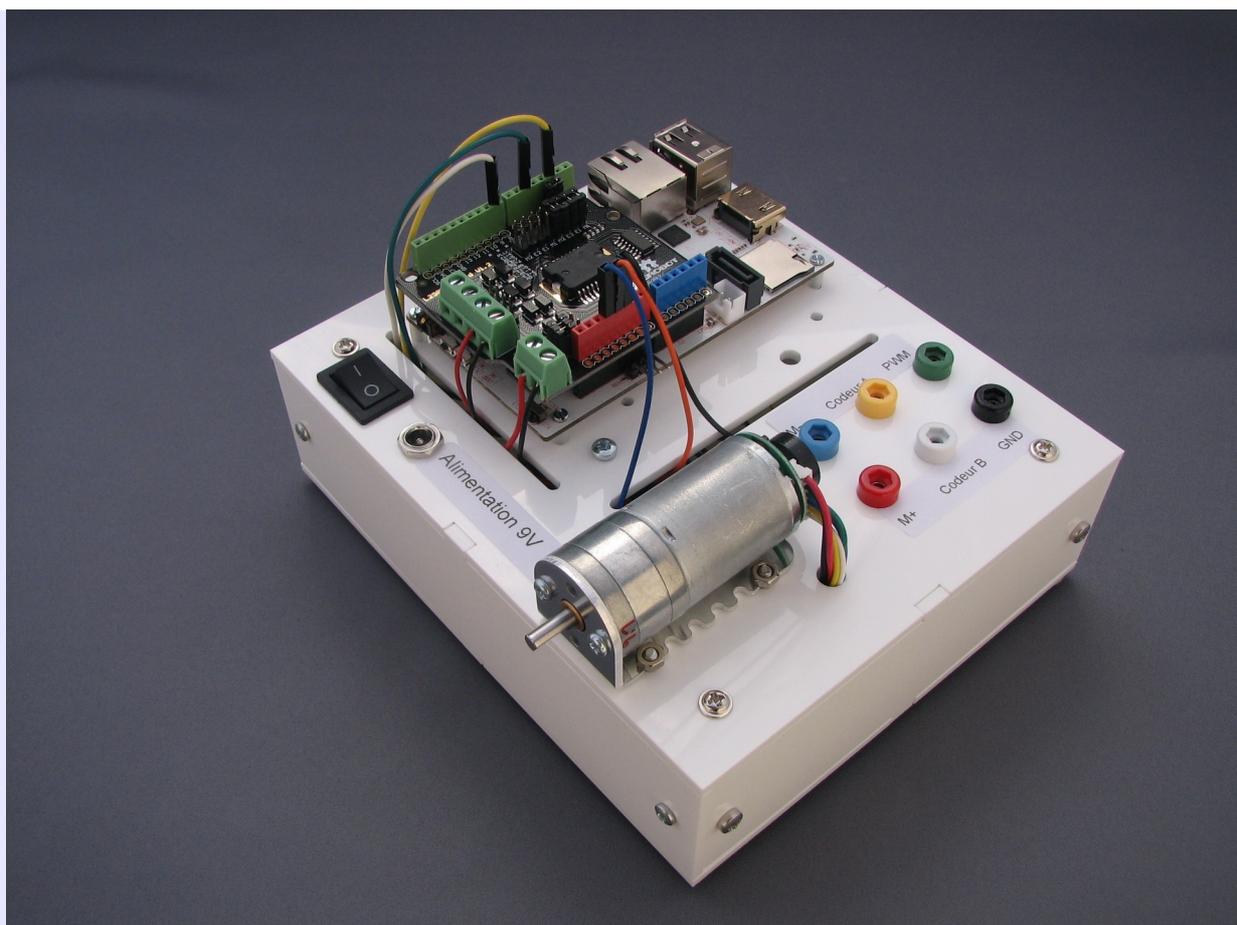




ENSEMBLE « COMMANDE DE MOTEUR À COURANT CONTINU » PCDUINO NANO

DOCUMENTATION COMPLÈTE



Date de dernière mise à jour : 02/12/2016

Table des matières

1 - Introduction.....	<u>3</u>
2 - Matériel inclus.....	<u>3</u>
3 - Conformité.....	<u>3</u>
4 - Mise en œuvre de l'ensemble.....	<u>4</u>
4.1 - Opérations préalables.....	<u>4</u>
4.2 - Utilisation standard.....	<u>4</u>
4.3 - Précautions d'emploi.....	<u>4</u>
4.3.1 - Connexions d'alimentation sur la carte de commande moteur.....	<u>4</u>
4.3.2 - Tension d'alimentation.....	<u>5</u>
4.3.3 - Utilisation.....	<u>5</u>
5 - Expériences de commande de moteur à courant continu.....	<u>6</u>
5.1 - Caractéristiques du moteur à courant continu avec codeur incrémental associé.....	<u>6</u>
5.1.1 - Calcul de la vitesse de rotation du moteur.....	<u>9</u>
5.1.2 - Comptage du nombre d'impulsions.....	<u>10</u>
5.2 - Commande en tension.....	<u>11</u>
5.3 - Asservissement de vitesse.....	<u>14</u>
6 - Accès aux programmes Python.....	<u>18</u>
7 - Important.....	<u>19</u>

1 - Introduction

Cet ensemble permet de réaliser différentes expériences sur la base d'un moteur à courant continu avec codeur incrémental associé, commandé par une carte pcDuino Nano.

Cette documentation, ainsi que les programmes associés sont téléchargeables sur notre site Web à l'adresse suivante:

<http://www.3sigma.fr/telechargements>

2 - Matériel inclus

Cet ensemble est livré monté et **fonctionnel (testé par nos soins avant la livraison)**. Il est composé des éléments suivants:

- un boîtier en plexiglas blanc, avec connecteur d'alimentation 5.5mm x 2.1mm, bouton marche-arrêt et douilles 2mm pour la mesure à l'oscilloscope de différents signaux.

Repérage des douilles 2mm:

Rouge	Bleu	Jaune	Blanc	Vert	Noir
+ moteur	- moteur	Voie A codeur	Voie B codeur	Signal PWM	Masse

- une carte pcDuino Nano
- un moteur à courant continu 6V, rapport de réduction 9.7:1, avec codeur incrémental 48 CPR. Le moteur est assemblé sur le boîtier par l'intermédiaire d'un support de montage en équerre
- 1 adaptateur USB / Ethernet permettant de communiquer directement entre un ordinateur (connexion USB) et le pcDuino (connexion Ethernet)
- 1 câble Ethernet
- 1 alimentation 9V, 1A avec connecteur d'alimentation 5.5mm x 2.1mm

3 - Conformité

L'ensemble « Commande de moteur à courant continu », **dans sa configuration livrée aux clients**, est conforme à la directive 1999/EC.

4 - Mise en œuvre de l'ensemble

4.1 - Opérations préalables

Attention : avant la première utilisation du système, vous devez installer sur votre ordinateur le pilote de l'adaptateur USB / Ethernet fourni avec le système. Ce pilote se trouve sur le CDROM livré avec l'adaptateur.

4.2 - Utilisation standard

La mise en œuvre de l'ensemble « Commande de moteur à courant continu » est très simple:

- brancher l'alimentation 9V fournie sur le connecteur jack du système
- connecter l'ordinateur au système pcDuino en utilisant le câble Ethernet et l'adaptateur USB – Ethernet (USB côté ordinateur)
- commuter l'interrupteur sur la position « I »
- attendre environ une minute le démarrage de la carte pcDuino
- exécuter l'un des logiciels de pilotage (commande en tension ou asservissement en vitesse, voir plus loin)

4.3 - Précautions d'emploi

Nous insistons sur le fait que cet ensemble est un matériel de développement qui nécessite un certain nombre de précautions d'emploi.

4.3.1 - Connexions d'alimentation sur la carte de commande moteur

Il est impératif de faire très attention aux connexions de l'alimentation du shield de commande moteur car celui-ci n'est pas protégé contre les inversions de polarité. Une erreur de connexion sur les bornes d'alimentation risque d'entraîner la destruction du sous-ensemble de gestion d'alimentation de la carte et de rendre celle-ci inutilisable. L'ensemble « Commande de moteur à courant continu » étant livré connecté et fonctionnel, il est préférable de ne pas modifier les branchements sur les connecteurs d'alimentation.

4.3.2 - Tension d'alimentation

Cet ensemble est prévu pour fonctionner avec l'alimentation 9V, 1A fournie. Le moteur a une tension nominale de 6V et bien qu'il supporte sans problème des tensions jusqu'à 9V, les programmes disponibles sur la carte pcDuino (ainsi que tous les programmes téléchargeables sur notre site) empêchent que la tension à ses bornes dépasse la valeur de 6V.

Il est possible d'utiliser un autre bloc d'alimentation à condition de bien respecter les points suivants:

- la tension ne doit pas dépasser 9V. Il est cependant recommandé de mettre éventuellement les programmes de la carte pcDuino à jour si ceux-ci sont basés sur une valeur de tension différente
- la polarité doit être « positif au centre du connecteur »
- l'alimentation doit pouvoir fournir un courant suffisant, de préférence supérieur ou égal à 1A

Notez que le système intègre un régulateur de tension 5V pour alimenter la carte pcDuino via la broche 5V du shield de commande moteur.

4.3.3 - Utilisation

Il est fortement déconseillé de faire des expériences de fonctionnement « rotor bloqué » avec une tension d'alimentation du moteur trop élevée. Ce type d'expérience peut générer des courants trop forts qui réduisent la durée des vies des éléments.

5 - Expériences de commande de moteur à courant continu

Cet ensemble permet de réaliser différentes expériences de commande de moteur à courant continu.

5.1 - Caractéristiques du moteur à courant continu avec codeur incrémental associé

L'expérience de commande de moteur électrique embarque un moteur à courant continu 6V, de rapport de réduction 9.7:1, avec codeur incrémental 48 CPR (Counts Per Revolution).

Ses équations sont les suivantes:

$$\frac{d}{dt} \omega_m(t) = \frac{\text{ratio} K i_m(t) - d \omega_m(t)}{J \cdot \text{ratio}^2}$$

$$\frac{d}{dt} i_m(t) = \frac{V(t) - R i_m(t) - K \cdot \text{ratio} \omega_m(t)}{L}$$

Avec :

- R : résistance électrique interne: 3.0 Ohms
- L : inductance des enroulements: 3.0 mH
- J : moment d'inertie du rotor: $3 \cdot 10^{-6}$ kg.m²
- K : constante de couple = constante de fem: 0.01 N.m/A
- d : coefficient de frottement visqueux: 0.0025 N.m.s/rad
- ω_m : vitesse de rotation de l'arbre de sortie du réducteur (rad/s)
- i_m : courant dans le moteur (A)
- V : tension d'alimentation (V)

Ces paramètres ont été identifiés à partir d'un essai de réponse du moteur à un échelon de tension.

La fonction de transfert entre l'entrée $V(t)$ et la sortie $\omega_m(t)$ est la suivante :

$$\frac{K \text{ ratio}}{J L \text{ ratio}^2 s^2 + (J R \text{ ratio}^2 + L d) s + K^2 \text{ ratio}^2 + R d}$$

La fonction de transfert entre l'entrée $V(t)$ et la sortie $i_m(t)$ est la suivante :

$$\frac{J \text{ ratio}^2 s + d}{J L \text{ ratio}^2 s^2 + (J R \text{ ratio}^2 + L d) s + K^2 \text{ ratio}^2 + R d}$$

Noter que les valeurs numériques données ci-dessus concernent l'identification du moteur avec son driver, l'influence de ce dernier étant importante et devant être pris en compte dans l'asservissement global du système.

Le brochage du moteur (couleur des fils) est donné dans le tableau ci-dessous:

Rouge	Noir	Bleu	Vert	Jaune	Blanc
+ moteur	- moteur	5V codeur	Masse codeur	Voie A codeur	Voie B codeur

ATTENTION !

Ne pas confondre la couleur des fils du moteur et la couleur des douilles 2mm.

5.1.1 - Calcul de la vitesse de rotation du moteur

Le codeur incrémental fournit deux signaux carrés en quadrature, comme sur la capture ci-dessous:



Ces deux signaux permettent de mesurer à la fois la vitesse et le sens de rotation. La mesure de la vitesse se fait simplement en comptant le nombre d'impulsions pendant un temps fixe. Les données du problème sont les suivantes :

- Le codeur est fixé à l'arbre moteur et non pas à l'arbre de sortie du réducteur (celui utilisé pour l'entraînement). Le rapport de réduction étant 9.7:1, l'arbre moteur fait 9.7 tours lorsque l'arbre « principal » en fait 1
- Le codeur génère 48 impulsions à chaque fois qu'il fait un tour
- La cadence d'échantillonnage utilisée pour l'asservissement sera de 0.01 s

Par conséquent, lorsque l'arbre principal fait un tour, le codeur génère :
 $9.7 * 48 \approx 465$ impulsions.

Si N est le nombre d'impulsions comptées en 0.01 s, la vitesse est (en rad/s, l'unité standard, sachant qu'un tour fait $2*\pi$ radians) :

$$2*\pi*N/(0.01*465)$$

ATTENTION !

Bien que le codeur soit placé sur l'arbre moteur, le calcul ci-dessus donne la vitesse en sortie du réducteur.

Un point très important concerne la résolution de la mesure, c'est-à-dire la plus petite valeur qu'il est possible de calculer. La formule est la suivante (en rad/s) :

$$2*\pi/(Ts*CPR*ratio)$$

avec :

- Ts : cadence d'échantillonnage
- CPR : nombre d'impulsions par tour du codeur
- ratio : rapport de réduction du moteur

Dans notre cas de figure, la résolution est la suivante

$$2*\pi/(0.01*465) = 1.35 \text{ rad/s}$$

5.1.2 - Comptage du nombre d'impulsions

Compter le nombre d'impulsions du codeur revient à compter le nombre de fronts montants et descendants des signaux jaune et bleu représentés sur l'image ci-dessus. Pour ce faire, la seule méthode viable consiste à brancher les deux signaux (les fils jaune et blanc sur le codeur utilisé) sur deux entrées « interruption » de la carte pcDuino. Les deux autres fils (bleu et vert) seront respectivement branchés sur le 3.3 V et sur la masse du pcDuino.

Sur une carte pcDuino, il y a deux lignes d'interruption (numérotées 0 et 1), qui correspondent aux broches digitales 2 et 3. L'intérêt d'une ligne d'interruption est qu'elle permet, comme son nom l'indique, d'interrompre le déroulement des calculs sur le processeur pour effectuer un traitement spécifique, en l'occurrence la mise à jour du compteur d'impulsions, avant de rendre la main à la boucle principale.

La seule « difficulté » est de savoir s'il faut incrémenter ou décrémenter le compteur dans le traitement de l'interruption. Il suffit pour cela d'observer les courbes ci-dessus, obtenues alors que le moteur tourne dans le sens positif. On constate que:

- Lorsque la voie A (en jaune) passe au niveau haut, la voie B (en bleu) est au niveau bas
- Lorsque la voie A passe au niveau bas, la voie B est au niveau haut

Quand le moteur tourne dans le sens positif, lors d'une interruption sur la voie A, les niveaux de A et B sont donc inversés.

En ce qui concerne l'interruption liée à la voie B, c'est l'inverse :

- Lorsque la voie B passe au niveau haut, la voie A est au niveau haut
- Lorsque la voie B passe au niveau bas, la voie A est au niveau bas

5.2 - Commande en tension

Cette expérience permet de changer la vitesse de rotation du moteur en appliquant une tension variable par l'intermédiaire d'une application qui s'exécute sur votre ordinateur.

Le programme Python gérant cette expérience se nomme `CommandeEnTension.py`. Il est situé dans le répertoire `/root/programmes_python`.

Il comporte de nombreux commentaires permettant de comprendre facilement son fonctionnement.

Le principe de pilotage du moteur consiste à envoyer des signaux PWM sur le shield de commande moteur connecté à la carte pcDuino afin de faire varier la tension d'alimentation du moteur. Cette tension peut être modifiée interactivement via une application qui s'exécute sur votre ordinateur. Elle peut se télécharger à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

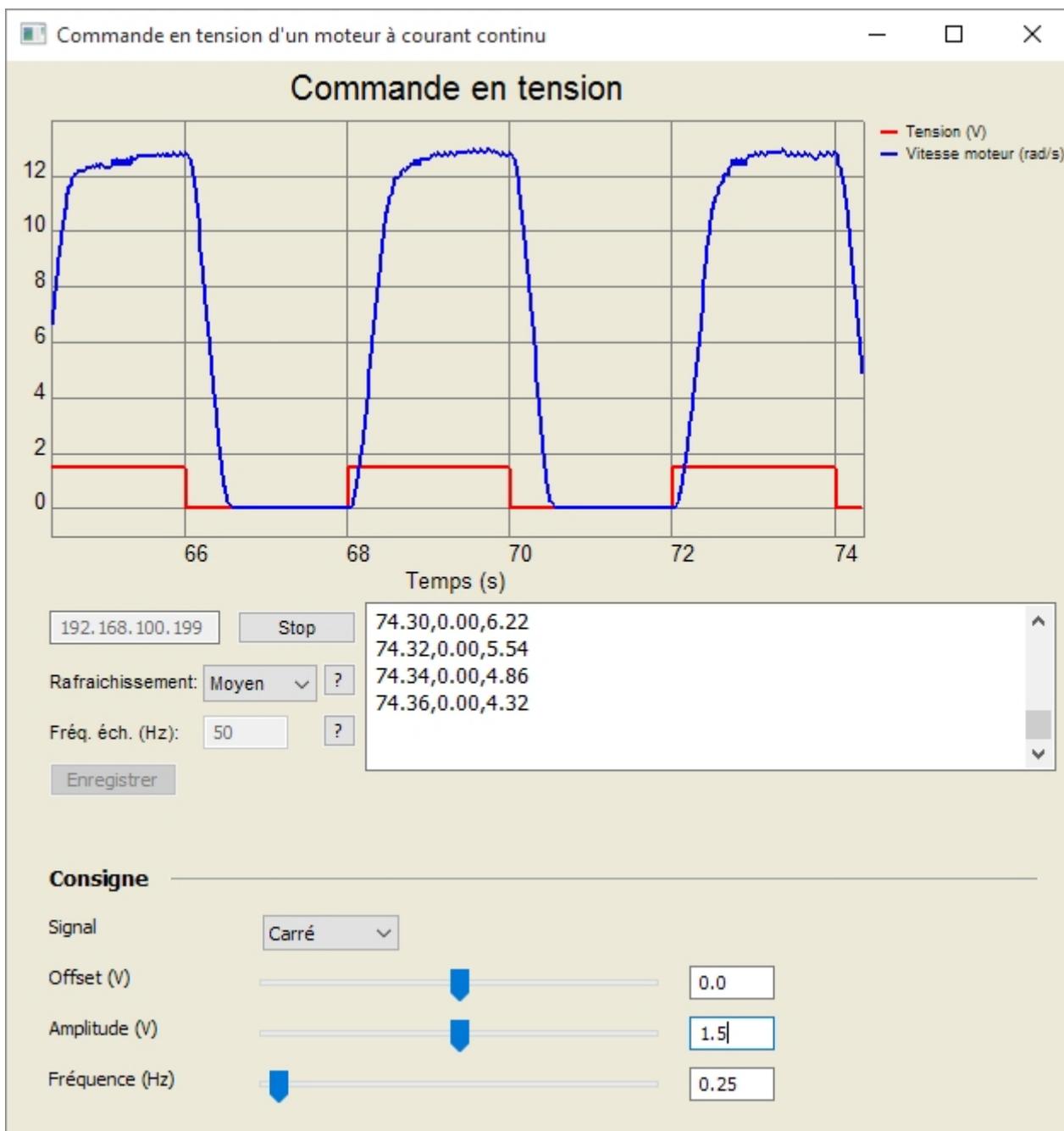
Son nom est de la forme `InterfaceReseauCommandeMoteurEnTension_x.y.zip` (x.y correspond au numéro de version du programme).

Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur `InterfaceReseauCommandeMoteurEnTension.exe`.

Pour piloter la tension d'alimentation (et donc la vitesse, mais sans asservissement de cette dernière) d moteur depuis votre ordinateur, il vous suffit de suivre les étapes suivantes :

- brancher l'alimentation 9V fournie sur le connecteur jack du système
- connecter l'ordinateur au système pcDuino en utilisant le câble Ethernet et l'adaptateur USB – Ethernet (USB côté ordinateur)
- commuter l'interrupteur sur la position « I »
- attendre environ une minute le démarrage de la carte pcDuino
- lancer l'application « `InterfaceReseauCommandeMoteurEnTension` »
- entrer l'adresse IP du système (192.168.100.199) dans la zone correspondante et cliquer sur « Connexion ». L'application démarre alors à distance le programme Python de commande du moteur sur la carte pcDuino. Vous pouvez alors interagir avec l'interface graphique pour faire vos expérimentations

Voici une capture d'écran de l'interface de l'application de pilotage:



Les différents éléments sont les suivants (de haut en bas)

- courbes: l'interface permet de visualiser
 - la tension de commande (V, en rouge)
 - la vitesse mesurée (rad/s, en bleu)
- zone de sélection de l'adresse IP: entrer l'adresse IP de votre système et cliquer sur le bouton « Connexion ». Vous verrez alors les courbes se mettre en jour automatiquement et défiler des valeurs numériques dans la zone d'affichage à droite
- « Rafraîchissement »: ceci pilote la fréquence de rafraîchissement des courbes. En fonction de la vitesse de votre ordinateur, vous pouvez choisir parmi les 4 valeurs « Minimum », « Lent », « Moyen » et « Rapide ». Plus la fréquence de rafraîchissement est élevée, moins le tracé est saccadé. Mais si votre ordinateur n'est pas très rapide, vous risquez d'observer un retard entre les consignes et l'affichage. Dans ce cas, il faut choisir un rafraîchissement plus lent
- « Fréq. éch. (Hz) »: ce paramètre correspond à la fréquence d'échantillonnage des mesures dans le programme `CommandeEnTension.py` (20 ms par défaut)
- Bouton « Enregistrer »: il permet d'enregistrer les valeurs numériques affichées dans la zone de « log » vers un fichier texte pour une utilisation ultérieure avec n'importe quel logiciel permettant de lire des données séparées par des virgules dans un fichier texte.
Attention : ce bouton n'est actif qu'après une séquence de mesure. Il est grisé le reste du temps (au démarrage du programme et pendant une séquence de mesure)
- Zone « Consigne »: elle permet de modifier la consigne de tension avec les curseurs. La vitesse du moteur varie alors. Vous pouvez la visualiser, ainsi que la consigne et la tension de commande, sur le graphique qui s'affiche en temps-réel

IMPORTANT !

Si vous comparez la vitesse de rotation obtenue sur votre ensemble avec celle affichée sur la capture d'écran au début de ce paragraphe, vous obtiendrez probablement une valeur différente, même si la tension de commande est la même : c'est normal, ceci est dû à la disparité des caractéristiques des moteurs à courant continu.

D'un point de vue pédagogique, ce point permet de souligner la nécessité de réaliser un asservissement de vitesse : si l'on souhaite une vitesse de rotation précise, on ne peut pas se contenter d'une commande en boucle ouverte.

5.3 - Asservissement de vitesse

Cette expérience permet d'asservir la vitesse de rotation du moteur en appliquant une consigne de vitesse par l'intermédiaire d'une application qui s'exécute sur votre ordinateur.

Le programme Python gérant cette expérience se nomme AsservissementVitesse.py. Il est situé dans le répertoire /root/programmes_python.

Il comporte de nombreux commentaires permettant de comprendre facilement son fonctionnement.

Le principe de pilotage du moteur consiste à calculer, grâce à un régulateur de type PID, la tension de commande à appliquer au moteur (via commande PWM du shield de commande moteur connecté sur la carte pcDuino) pour qu'il suive la consigne de vitesse spécifiée.

Notez que dans ce programme, la vitesse mesurée est en fait la moyenne glissante des 40 derniers échantillons de mesure de vitesse instantanée, ce qui permet d'avoir une mesure plus lisse. En effet, la résolution de la mesure instantanée est la suivante:

$$2*\pi/(Ts*CPR*ratio)$$

avec :

- Ts : cadence d'échantillonnage (0.01 s)
- CPR : nombre d'impulsions par tour du codeur (48)
- ratio : rapport de réduction du moteur (9.7)

Ceci donne une résolution de $2*\pi/(0.01*465) = 1.35$ rad/s. Le moyennage permet d'améliorer cette valeur.

La consigne de vitesse peut se fixer interactivement via une application qui s'exécute sur votre ordinateur. Elle peut se télécharger à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

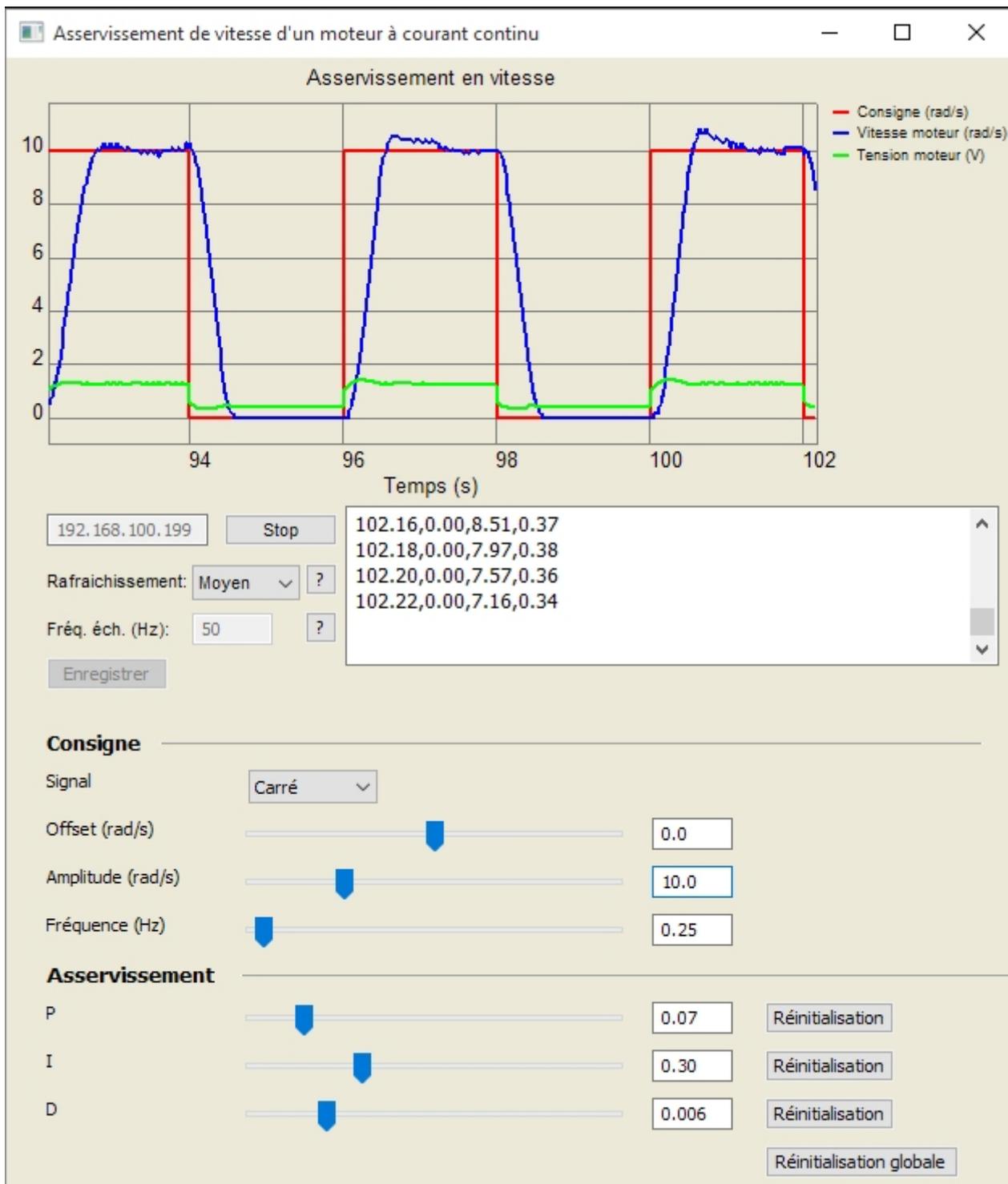
Son nom est de la forme InterfaceReseauAsservissementMoteurEnVitesse_x.y.zip (x.y correspond au numéro de version du programme).

Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur InterfaceReseauAsservissementMoteurEnVitesse.exe.

Pour changer la consigne de vitesse du moteur depuis votre ordinateur, il vous suffit de suivre les étapes suivantes (remarque: n'exécutez pas de nouveau celles que vous avez déjà effectuées):

- brancher l'alimentation 9V fournie sur le connecteur jack du système
- connecter l'ordinateur au système pcDuino en utilisant le câble Ethernet et l'adaptateur USB – Ethernet (USB côté ordinateur)
- commuter l'interrupteur sur la position « I »
- attendre environ une minute le démarrage de la carte pcDuino
- lancer l'application « InterfaceReseauAsservissementMoteurEnVitesse »
- entrer l'adresse IP du système (192.168.100.199) dans la zone correspondante et cliquer sur « Connexion ». L'application démarre alors à distance le programme Python d'asservissement du moteur sur la carte pcDuino. Vous pouvez alors interagir avec l'interface graphique pour faire vos expérimentations

Voici une capture d'écran de l'interface de l'application de pilotage:

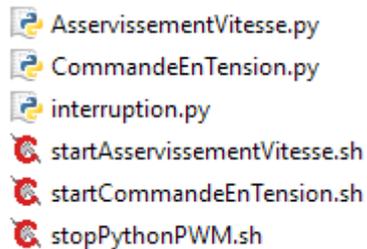


Les différents éléments sont les suivants (de haut en bas)

- courbes: l'interface permet de visualiser
 - la consigne de vitesse (rad/s, en rouge)
 - la vitesse mesurée (rad/s, en bleu)
 - la tension de commande (V, en vert)
- zone de sélection de l'adresse IP: entrer l'adresse IP de votre système et cliquer sur le bouton « Connexion ». Vous verrez alors les courbes se mettre en jour automatiquement et défiler des valeurs numériques dans la zone d'affichage à droite
- « Rafraîchissement »: ceci pilote la fréquence de rafraîchissement des courbes. En fonction de la vitesse de votre ordinateur, vous pouvez choisir parmi les 4 valeurs « Minimum », « Lent », « Moyen » et « Rapide ». Plus la fréquence de rafraîchissement est élevée, moins le tracé est saccadé. Mais si votre ordinateur n'est pas très rapide, vous risquez d'observer un retard entre les consignes et l'affichage. Dans ce cas, il faut choisir un rafraîchissement plus lent
- « Fréq. éch. (Hz) »: ce paramètre correspond à la fréquence d'échantillonnage des mesures dans le programme AsservissementVitesse.py.
- Bouton « Enregistrer »: il permet d'enregistrer les valeurs numériques affichées dans la zone de « log » vers un fichier texte pour une utilisation ultérieure avec n'importe quel logiciel permettant de lire des données séparées par des virgules dans un fichier texte.
Attention : ce bouton n'est actif qu'après une séquence de mesure. Il est grisé le reste du temps (au démarrage du programme et pendant une séquence de mesure)
- Zone « Consigne »: elle permet de modifier la consigne de vitesse avec les curseurs. La vitesse du moteur varie alors. Vous pouvez la visualiser, ainsi que la consigne et la tension de commande, sur le graphique qui s'affiche en temps-réel
- Zone « Asservissement »: elle permet de modifier les gains du régulateur PID pendant le fonctionnement du système. Cela permet de voir l'influence de ces gains sur les performances de l'asservissement et de régler précisément ce dernier.

6 - Accès aux programmes Python

Vous pouvez accéder aux programmes Python de gestion du moteur sur la carte pcDuino dans le répertoire `/root/programmes_python` :



Ces programmes sont librement consultables et modifiables en vous connectant en SSH à la carte pcDuino avec les informations suivantes :

- adresse IP : 192.168.100.199
- login : root
- mot de passe : pcdduino

Nous vous recommandons l'utilisation du logiciel WinSCP (<https://winscp.net/eng/download.php>) qui permet de naviguer très aisément dans l'arborescence des répertoires de la carte pcDuino et de faire facilement des transferts de fichiers entre votre ordinateur hôte et la pcDuino.

7 - Important

Cet ensemble est un produit « vivant » en constant développement pour améliorer ou lui ajouter de nouvelles fonctionnalités. Si vous avez des idées ou des besoins pour des développements spécifiques, n'hésitez pas à nous contacter (info@3sigma.fr).

Ne restez jamais bloqué sans nous contacter !

Pour tout problème ou toute requête, contactez-nous à l'adresse support@3sigma.fr.