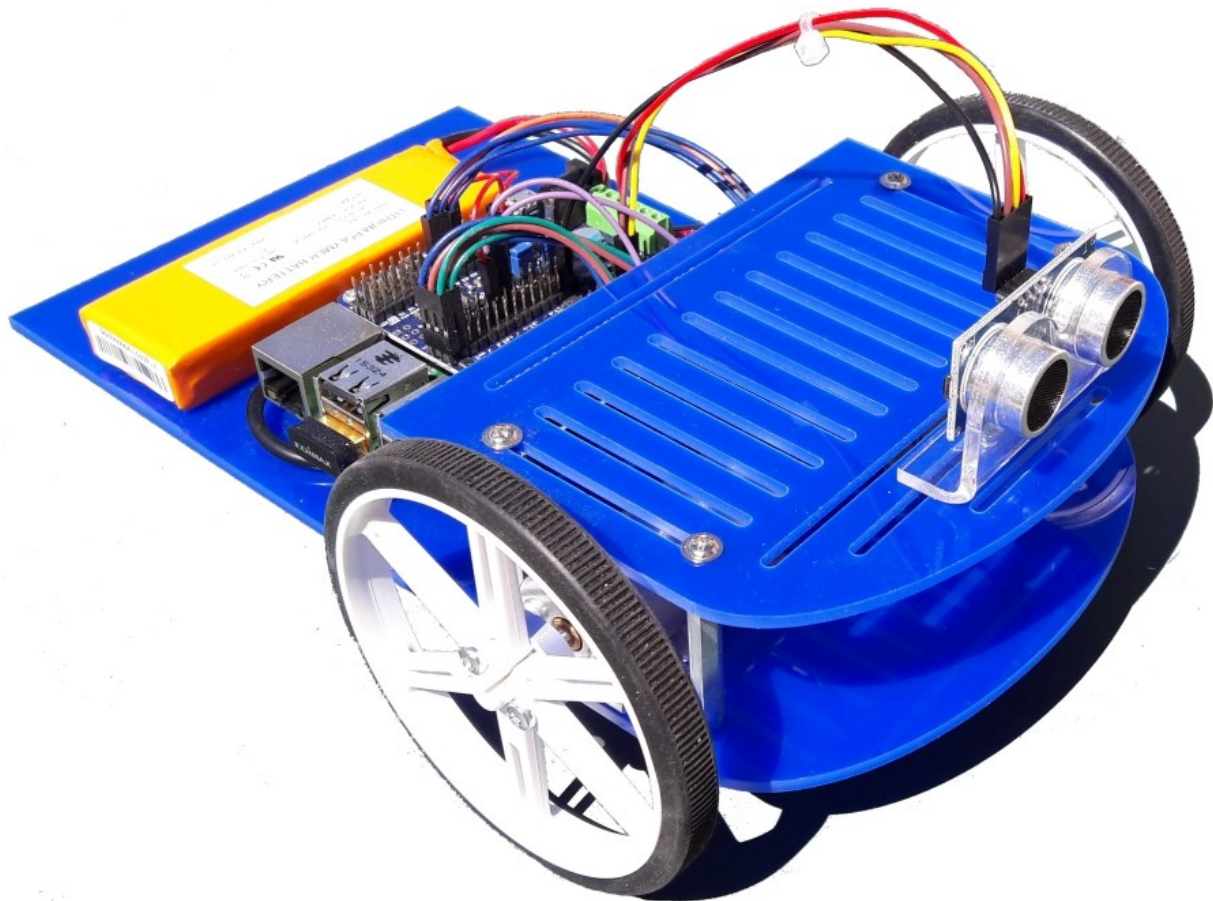




X-BOT RASPBERRY C / PYTHON

DOCUMENTATION COMPLÈTE



Date de dernière mise à jour : 27/04/2017

Table des matières

1 - Introduction.....	<u>4</u>
2 - Matériel inclus.....	<u>4</u>
3 - Conformité.....	<u>4</u>
4 - Installation de l'IDE Arduino.....	<u>5</u>
4.1 - Installation principale.....	<u>5</u>
4.2 - Installation du support de la carte A-Star.....	<u>5</u>
4.3 - Installation de la bibliothèque complémentaire FlexiTimer2.....	<u>6</u>
4.4 - Téléchargement d'un programme sur la carte A-Star.....	<u>6</u>
5 - Précautions d'emploi.....	<u>7</u>
5.1 - Connexions d'alimentation sur la carte de commande moteur.....	<u>7</u>
5.2 - Recharge de la batterie.....	<u>7</u>
5.3 - Précautions d'utilisation.....	<u>8</u>
6 - Logiciels.....	<u>9</u>
6.1 - Programmes C/C++.....	<u>9</u>
6.2 - Programmes Python.....	<u>10</u>
6.3 - Application Web de pilotage.....	<u>10</u>
7 - Utilisation.....	<u>11</u>
7.1 - Mise en route.....	<u>11</u>
7.2 - Pilotage par ordinateur.....	<u>12</u>
7.3 - Pilotage par smartphone ou tablette.....	<u>15</u>
7.4 - Suivi de ligne.....	<u>18</u>
7.5 - Evitement d'obstacle.....	<u>20</u>
8 - Asservissements et expériences additionnelles.....	<u>21</u>
8.1 - Caractéristiques du moteur à courant continu avec codeur incrémental associé.....	<u>21</u>
8.1.1 - Calcul de la vitesse de rotation du moteur.....	<u>22</u>
8.1.2 - Comptage du nombre d'impulsions.....	<u>24</u>
8.2 - Modèle du robot.....	<u>24</u>
8.3 - Commande des moteurs en tension (en Python).....	<u>26</u>
8.4 - Asservissement des moteurs en vitesse (en Python).....	<u>29</u>
8.5 - Commande des moteurs en tension (en C/C++).....	<u>34</u>
8.6 - Asservissement des moteurs en vitesse (en C/C++).....	<u>37</u>
9 - Autre documentation disponible.....	<u>41</u>

10 - Important.....[42](#)

1 - Introduction

X-Bot Raspberry C / Python est un robot ouvert et open-source, un concentré de technologie vous permettant de faire de nombreuses expériences. Il est basé sur la carte Raspberry Pi associée à une carte de commande moteur « Pololu A-star », différents capteurs et deux moteurs électriques permettant de le mettre en mouvement.

Des ressources associées à X-Bot Raspberry C / Python sont téléchargeables sur notre site Web à l'adresse suivante:

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

2 - Matériel inclus

X-Bot Raspberry Python est livré monté et **fonctionnel (programmé et testé par nos soins avant la livraison)**. Il est composé des éléments suivants:

- Le robot lui-même
- 1 chargeur de batterie LiPo 8.4V, 1A
- 1 clé allen pour visser ou dévisser les roues

3 - Conformité

Le robot X-Bot Raspberry Python, **dans sa configuration livrée aux clients**, est conforme à la directive 1999/EC.



4 - Installation de l'IDE Arduino

Cette version d'X-Bot intègre une carte A-Star, compatible Arduino. A la livraison, cette carte est programmée pour effectuer les opérations de bas niveau (calcul de la vitesse de rotation des moteurs et commande de ces derniers) en relation avec la Raspberry Pi, qui réalise les asservissements de haut niveau.

Cependant, il est également possible de programmer l'A-Star pour que ce soit elle qui effectue la totalité du contrôle du robot. Ceci nécessite de télécharger un nouveau programme et pour ce faire, vous devez au préalable installer l'IDE Arduino et quelques bibliothèques additionnelles.

4.1 - Installation principale

Télécharger et installer l'IDE Arduino (<http://arduino.cc/en/Main/Software>). Il suffit de décompresser l'archive téléchargée dans le répertoire de votre choix.

4.2 - Installation du support de la carte A-Star

Il faut effectuer les opérations suivantes pour ajouter le support de la carte A-Star dans l'IDE Arduino :

- Télécharger et décompresser l'archive suivante dans le répertoire de votre choix :
https://www.pololu.com/file/download/a-star-2.0.0.zip?file_id=0J743
- Ouvrir le sous-répertoire « drivers ». Faire un clic bouton droit sur le fichier a-star.inf et sélectionner « Installer »
- Repérer l'emplacement de votre « Répertoire de croquis » en sélectionnant Fichiers → Préférences dans l'IDE Arduino. Ce répertoire est indiqué à l' « Emplacement du carnet de croquis »
- Copier dans ce répertoire de croquis le répertoire nommé « pololu », que vous trouverez dans l'archive précédemment téléchargée
- Redémarrer l'IDE Arduino
- Dans le menu Outils → Type de carte, choisir « Pololu A-Star 32u4 »
- Dans le menu Croquis → Inclure ou bibliothèque, sélectionner « Gérer les bibliothèques »
- Chercher AStar32U4
- Sélectionner l'entrée AStar32U4 trouvée et cliquer sur « Installer »
- Redémarrer l'IDE Arduino

4.3 - Installation de la bibliothèque complémentaire FlexiTimer2

Cette bibliothèque permet d'exécuter à cadence fixe une partie du programme Arduino.

Vous pouvez la télécharger à l'adresse suivante: <http://www.3sigma.fr/telechargements/FlexiTimer2.zip>.

Une fois téléchargée, décompressez-là dans le répertoire des librairies de votre installation Arduino (typiquement, Documents\Arduino\libraries).

ATTENTION !

L'environnement Arduino doit être redémarré après l'installation des bibliothèques complémentaires.

4.4 - Téléchargement d'un programme sur la carte A-Star

ATTENTION !

Pour télécharger un programme sur la carte A-Star, il est impératif de basculer au préalable le bouton marche-arrêt du robot sur la position « I ». Geeros doit par ailleurs être posé sur l'arrière.

La procédure à suivre pour la programmation est la suivante:

- Connecter le câble USB reliant l'ordinateur et la carte A-Star
- Mettre le robot sous tension en appuyant sur le bouton marche-arrêt
- Lancer le téléchargement
- Eteindre et rallumer le robot

5 - Précautions d'emploi

Nous insistons sur le fait que X-Bot Raspberry Python est un robot d'étude et de développement qui nécessite un certain nombre de précautions d'emploi.

5.1 - 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 protégée 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. Le robot étant livré connecté et fonctionnel, il est préférable de ne pas modifier les branchements sur les connecteurs d'alimentation.

5.2 - Recharge de la batterie

Le chargeur 8.4V, 1A fourni est destiné à recharger la batterie LiPo du robot en le connectant sur le connecteur jack situé à côté du bouton de marche-arrêt. Bien que cette batterie possède une protection intégrée contre les surcharges et les sous-charges, il n'est pas conseillé de la recharger avec une alimentation pouvant fournir un courant supérieur à 1A.

Si une autre alimentation devait être utilisée pour la recharger, il est important de vérifier que sa polarité est « positif au centre du connecteur ».

Il faut compter environ 2 heures pour une demi-recharge de la batterie et 8 heures pour une recharge complète.

5.3 - Précautions d'utilisation

X-Bot Raspberry Python doit être utilisé dans les conditions suivantes:

- Au sol. Ne jamais l'utiliser en hauteur (sur une table, par exemple) à cause des risques de chute
- A l'intérieur. Les sols extérieurs présentent souvent des aspérités et des obstacles surdimensionnés par rapport à la taille du robot. Par ailleurs, X-Bot Raspberry Python craint l'eau et l'humidité

X-Bot Raspberry Python est fourni avec différents programmes. Vous pouvez par exemple étudier la commande des moteurs électriques. Mais attention: 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 vie des éléments.

6 - Logiciels

X-Bot Raspberry C / Python associe deux systèmes « intelligents » :

- Une carte A-Star (compatible Arduino) à base de micro-contrôleur ATmega32u4
- Une carte Raspberry Pi

Les programmes en C/C++ (à la « mode » Arduino) sont destinés à être exécutés sur la carte A-Star alors que les programmes Python sont exécutés par la carte Raspberry Pi.

Les différentes fonctionnalités du robot (commande des moteurs en tension, asservissement des moteurs en vitesse ou contrôle total du robot) peuvent être gérées par l'une ou l'autre des deux cartes. Cependant, lorsque ce sont les programmes Python de la Raspberry Pi qui sont mis en œuvre, la carte A-Star reste sollicitée pour effectuer les opérations les plus directement liées au matériel, à savoir le comptage des impulsions des codeurs incrémentaux et la génération des PWM pour la commande des moteurs électriques. Il est indispensable de charger sur l'A-Star un programme, que nous appellerons par la suite « firmware », dès que l'on souhaite utiliser les programmes Python de la Raspberry Pi.

6.1 - Programmes C/C++

Les programmes C/C++ se téléchargent sur la page http://www.3sigma.fr/Telechargements-X_Bot.html#pcduino, section « Programmes de la carte A-Star ».

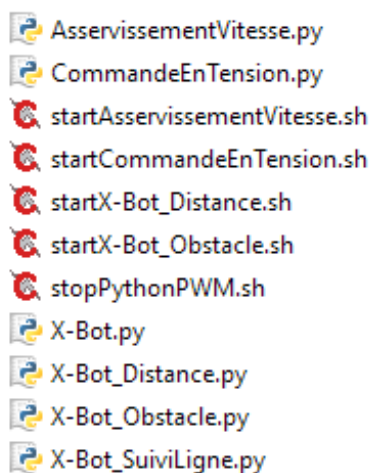
Ces programmes sont les suivants :

- Firmware de communication avec la carte Raspberry Pi : doit impérativement être chargé si on souhaite exécuter les programmes Python de la Raspberry Pi. En effet, ce firmware gère les entrées / sorties bas niveau et la communication avec la Raspberry Pi.
- Commande des moteurs en tension (exécutée uniquement par la carte A-Star)
- Asservissement des moteurs en vitesse (exécutée uniquement par la carte A-Star)

6.2 - Programmes Python

Un certain nombre de programmes Python sont fournis pour piloter le robot, faire du suivi de ligne, de l'évitement d'obstacles, du pilotage en tension des moteurs et de l'asservissement en vitesse de ces derniers.

Ces programmes se trouvent dans le répertoire `/root/programmes_python` de la carte Raspberry Pi :



6.3 - Application Web de pilotage

Le pilotage du robot via un ordinateur ou un smartphone est une application Web dont le code source se trouve dans le répertoire `/www` de la carte Raspberry Pi.

7 - Utilisation

7.1 - Mise en route

Les étapes à suivre pour utiliser rapidement X-Bot sont les suivantes :

- mettre le robot sous tension
- attendre environ une minute la création du point d'accès Wifi « X-Bot »
- se connecter en Wifi sur ce réseau « X-Bot ». Le mot de passe Wifi est « xbotraspberry »
- se connecter en SSH au robot à l'adresse 192.168.0.199 (login : « root », mot de passe : « raspberrypi »)
- terminer tout programme Python déjà en cours d'exécution en exécutant la commande suivante :
`/root/programmes_python/stopPythonPWM.sh`
- démarrer le programme Python de pilotage en exécutant la commande suivante :
`python /root/programmes_python/X-Bot.py`
- sur votre ordinateur, naviguer à l'adresse <http://192.168.0.199>
- choisir le pilotage par ordinateur : vous pouvez alors utiliser le joypad virtuel pour donner des consignes de mouvement au robot

7.2 - Pilotage par ordinateur

Pour pouvoir piloter X-Bot en Wifi grâce à votre ordinateur, les conditions suivantes doivent être respectées:

- Votre ordinateur doit posséder une interface Wifi. C'est le cas de la quasi-totalité des ordinateurs portables modernes ; si vous utilisez un ordinateur de bureau, vous devrez probablement connecter une « clé Wifi » à ce dernier pour lui apporter ce mode de communication
- Vous devez avoir rejoint le réseau Wifi « X-Bot » (voir section précédente)
- Vous devez utiliser un navigateur Web récent

Si toutes ces conditions sont remplies, la connexion de votre navigateur à l'adresse 192.168.0.199 conduira à la page suivante:



Trois choix sont alors possibles:

- Pilotage par ordinateur (voir ci-dessous)
- Pilotage par smartphone: si vous êtes connecté avec un ordinateur, cliquer sur cette icône ne vous apportera rien car votre ordinateur ne possède a priori pas d'accéléromètre (utilisé pour le pilotage avec les appareils mobiles). Cependant, si la détection automatique du type d'appareil ne fonctionne pas, l'application Web pourrait ne pas reconnaître votre smartphone ou tablette et faire pointer son navigateur sur ce tableau de bord. Le clic sur « Pilotage par smartphone » permet alors à votre appareil mobile d'accéder à l'application de pilotage qui lui est dédiée
- Aide: résumé des 2 points précédents pour ceux qui ne souhaitent pas lire la présente documentation

Pour piloter X-Bot avec votre ordinateur, vous devez bien sûr cliquer sur la première icône. Vous obtenez alors la page suivante:



Pilotage par ordinateur



Mouvement

Consigne de vitesse longitudinale: 0 cm/s

Consigne de vitesse de rotation: 0 deg/s

Asservissement de vitesse

P:  1.0

I:  1.0

Pour afficher dans cette zone les images prises par le robot, il vous suffit de brancher une Webcam compatible UVC sur le port USB de la carte de communication (entre les plateaux inférieur et supérieur).

Redémarrez ensuite le robot et rafraîchissez cette page dans votre navigateur.

www.geeros.com

Cette interface présente différents éléments

- Le joystick « Mouvement » permet de donner:
 - une consigne de vitesse de rotation uniquement, lorsque le centre de la boule jaune se trouve dans la moitié supérieure du disque gris
 - une consigne de vitesse de rotation et une consigne de vitesse de translation lorsque le centre de la boule jaune se trouve dans la moitié supérieure de l'anneau bleu
 - une consigne de vitesse de translation (négative) uniquement, lorsque le centre de la boule jaune se trouve dans la moitié inférieure de l'anneau bleu
 - aucune consigne lorsque le centre de la boule jaune se trouve dans la moitié inférieure du disque gris
- Les curseurs « P » et « I » permettent de modifier les paramètres du régulateur Proportionnel-Intégral de l'asservissement de vitesse du robot. Une diminution de « P » diminue la rapidité de l'asservissement. Une diminution de I dégrade la précision. Ces paramètres varient entre 0.5 et 2.
- A droite de l'écran, une zone est laissée libre dans le cas où vous rajouteriez une Webcam au robot. Dans ce cas, vous pourriez voir la retransmission vidéo en temps-réel de cette caméra.

7.3 - Pilotage par smartphone ou tablette

Pour pouvoir piloter X-Bot en Wifi grâce à votre smartphone ou votre tablette (nous utiliserons par la suite le nom générique « appareil »), les conditions suivantes doivent être respectées:

- Votre appareil doit posséder une interface Wifi
- Il doit posséder un accéléromètre
- Il doit être orienté en mode « paysage », comme ceci:



- Vous devez avoir rejoint le réseau Wifi « X-Bot » (voir section précédente)
 - sur iPhone, iPad et iPod: votre appareil doit être sous iOS 4.2 ou supérieur
 - sur Android: vous devez utiliser un navigateur Web récent
 - les autres systèmes n'ont pas encore été testés mais peuvent malgré tout fonctionner

Lorsque vous connectez le navigateur Web de votre appareil à l'adresse 192.168.0.199, vous obtenez tout d'abord la page suivante:

Pour afficher dans cette zone les images prises par le robot,
il vous suffit de brancher une Webcam compatible UVC
sur le port USB de la carte de communication
(entre les plateaux inférieur et supérieur).

Redémarrez ensuite le robot et rafraîchissez cette page dans votre navigateur.

Pilotage:

- Basculez votre appareil vers l'avant pour avancer
- Penchez-le à gauche ou à droite pour tourner

Pour démarrer, redressez l'appareil vers vous jusqu'à obtenir une valeur négative pour le nombre suivant:

Si rien ne s'affiche ou pour avoir de l'aide, [cliquer ici](#).

Consigne de vitesse
de translation: cm/s

Consigne de vitesse
de rotation: deg/s

Cette page permet de tester si votre appareil est compatible. Si tout se passe bien, vous devez voir, à droite de la zone vidéo (utile si vous rajoutez une Webcam à votre robot), un nombre changer en fonction de l'inclinaison de votre appareil. Ce nombre correspond à l'accélération de la pesanteur mesurée par votre appareil, normalisée entre 0 et 1. Si vous ne voyez pas ce nombre, cela signifie:

- que votre appareil ne possède pas d'accéléromètre
- ou bien qu'il n'est pas compatible avec cette application Web de pilotage (iPhone, iPad ou iPod avec iOS < 4.2, par exemple)
- ou bien que le navigateur utilisé n'est pas compatible

Dans ce cas, merci de nous contacter à l'adresse support@3sigma.fr en nous indiquant les caractéristiques précises du smartphone ou de la tablette que vous utilisez. Une solution personnalisée est peut-être possible.

Si vous voyez l'accélération évoluer en fonction de l'inclinaison de votre appareil, tout va bien. Avant d'accéder au pilotage à proprement parler, vous devez activer l'application. C'est très simple, il suffit de redresser votre appareil jusqu'à obtenir une accélération nulle. Une fois ceci fait, l'interface change légèrement.

L'accélération n'est plus visible, seules restent les consignes de vitesse de translation et de rotation (en plus de la transmission vidéo temps-réel de l'éventuelle Webcam, à gauche de l'écran). Piloter X-Bot est alors très facile:

- Lorsque vous basculez votre appareil vers l'avant, vous modifiez la consigne de vitesse de translation (entre 0 et 50 cm/s)
- Lorsque vous inclinez votre appareil sur la gauche ou sur la droite, vous modifiez la consigne de vitesse de rotation (entre -180 et 180 deg/s), comme avec un volant. Si l'inclinaison est trop forte à gauche ou à droite, l'application détecte que votre appareil est en mode portrait et annule les consignes

Remarque :

Dans ce mode de pilotage par smartphone ou tablette, il est impossible de faire reculer le robot car il est impossible de lui donner une consigne de vitesse négative.

7.4 - Suivi de ligne

Cette expérience permet au robot de suivre une ligne noire sur fond blanc à vitesse constante (les moteurs sont alimentés avec une tension de commande de 3V dans les phases de ligne droite).

Les conditions préalables au bon fonctionnement sont les suivantes:

- Les capteurs de suivi de lignes doivent être présents (ils sont facilement démontables...)
- mettre le robot sous tension
- attendre environ une minute la création du point d'accès Wifi « X-Bot »
- se connecter en Wifi sur ce réseau « X-Bot ». Le mot de passe Wifi est « xbotraspberry »
- se connecter en SSH au robot à l'adresse 192.168.0.199 (login : « root », mot de passe : « raspberrypi »)
- arrêter le programme en cours d'exécution:
`/root/programmes_python/stopPythonPWM.sh`
- démarrer le programme Python de suivi de ligne en exécutant la commande suivante :
`python /root/programmes_python/X-Bot_SuiviLigne.py`
- poser le robot sur le circuit, en essayant de préférence de positionner les capteurs au-dessus de la ligne noire
- le robot démarre quelques secondes plus tard et suit la ligne à une vitesse définie dans le programme Python pré-cité.

Il est important de conserver en mémoire les éléments susceptibles de jouer sur la détection de la ligne:

- Le contraste entre la ligne et le fond. Les meilleurs résultats sont obtenus en réalisant un circuit à partir de morceaux de lignes noires imprimées sur du papier blanc. Si vous avez un sol très clair, vous pouvez également réaliser un circuit directement par terre en utilisant du ruban adhésif pour tracer les lignes.
- La distance entre les capteurs et le sol: il faut trouver un bon compromis car plus la distance est faible, plus la ligne sera détectée correctement, mais plus les capteurs risqueront de rencontrer des obstacles. Par exemple, si un élément du circuit n'est pas très bien plaqué au sol, les capteurs du robot risquent de s'y accrocher. A la livraison du robot, la distance entre les capteurs et le sol est nominale et ne devrait pas être modifiée

Ces deux éléments ont une influence sur la capacité du robot à détecter la ligne sur le fond. Dans le programme X-Bot_SuiviLigne.py, c'est la variable « seuil » qui définit le seuil de détection. Lorsqu'un capteur est au dessus d'une ligne noire, la valeur remontée par le capteur est normalement supérieure à ce seuil (la valeur maximale étant 3300 mV). Lorsque le capteur est au-dessus d'un fond clair, sa valeur doit être inférieure au seuil (la valeur minimale étant 0 mV) mais en pratique, compte-tenu de l'ombre générée par le robot, les valeurs sont de l'ordre de 2000 mV.

Si le fond n'est pas assez clair ou si la ligne n'est pas assez noire, la valeur du seuil peut ne pas être optimale. Dans ce cas, il sera nécessaire de la changer de telle sorte que la relation suivante soit respectée :

valeur capteur sur fond clair < seuil < valeur capteur sur fond noir

Les caractéristiques nominales d'un circuit de suivi de ligne sont les suivantes:

- la largeur de la ligne devrait couvrir la largeur des 3 capteurs de suivi de ligne du robot
- le rayon de courbure ne doit pas être trop petit

7.5 - Evitement d'obstacle

Cette expérience utilise le capteur ultrasons HC-SR04 fixé sur le robot. Il réalise deux mouvements, en fonction de sa distance à l'obstacle :

- si la distance est supérieure à 40 cm, il avance en ligne droite à la vitesse de 20 cm/s
- sinon, il tourne sur la droite d'un quart de tour

Les conditions préalables au bon fonctionnement sont les suivantes:

- le capteur ultrason doit être présent (ils est facilement démontable et est absent dans les premières versions du robot)
- mettre le robot sous tension
- attendre environ une minute la création du point d'accès Wifi « X-Bot »
- se connecter en Wifi sur ce réseau « X-Bot ». Le mot de passe Wifi est « xbotraspberry »
- se connecter en SSH au robot à l'adresse 192.168.0.199 (login : « root », mot de passe : « raspberrypi »)
- démarrer le programme Python d'évitement d'obstacle en exécutant la commande suivante :
`/root/programmes_python/startX-Bot_Obstacle.sh`
- poser le robot par terre. Le mouvement débute 10 s après le démarrage du programme

Noter qu'à des fins de test de mesure de distance, vous pouvez utiliser X-Bot en mode pilotage classique avec affichage de la distance dans la fenêtre SSH. Pour ce faire, il faut exécuter la commande suivante :

`/root/programmes_python/startX-Bot_Distance.sh`

8 - Asservissements et expériences additionnelles

X-Bot Raspberry Python est un système pédagogique multi-expériences qui permet non seulement d'étudier de près le fonctionnement d'un robot mobile, mais également d'autres aspects plus spécifiques, comme la commande de moteurs électriques.

ATTENTION !

Veillez à sécuriser le robot lorsque vous réalisez les expériences décrites par la suite, pour qu'il ne risque pas de tomber d'une table, par exemple.

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

Ce robot embarque deux moteurs à courant continu 6V, de rapport de réduction 34:1, avec codeur incrémental 48 CPR (Counts Per Revolution).

Les équations d'un moteur 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 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 d'un moteur à un échelon de tension.

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

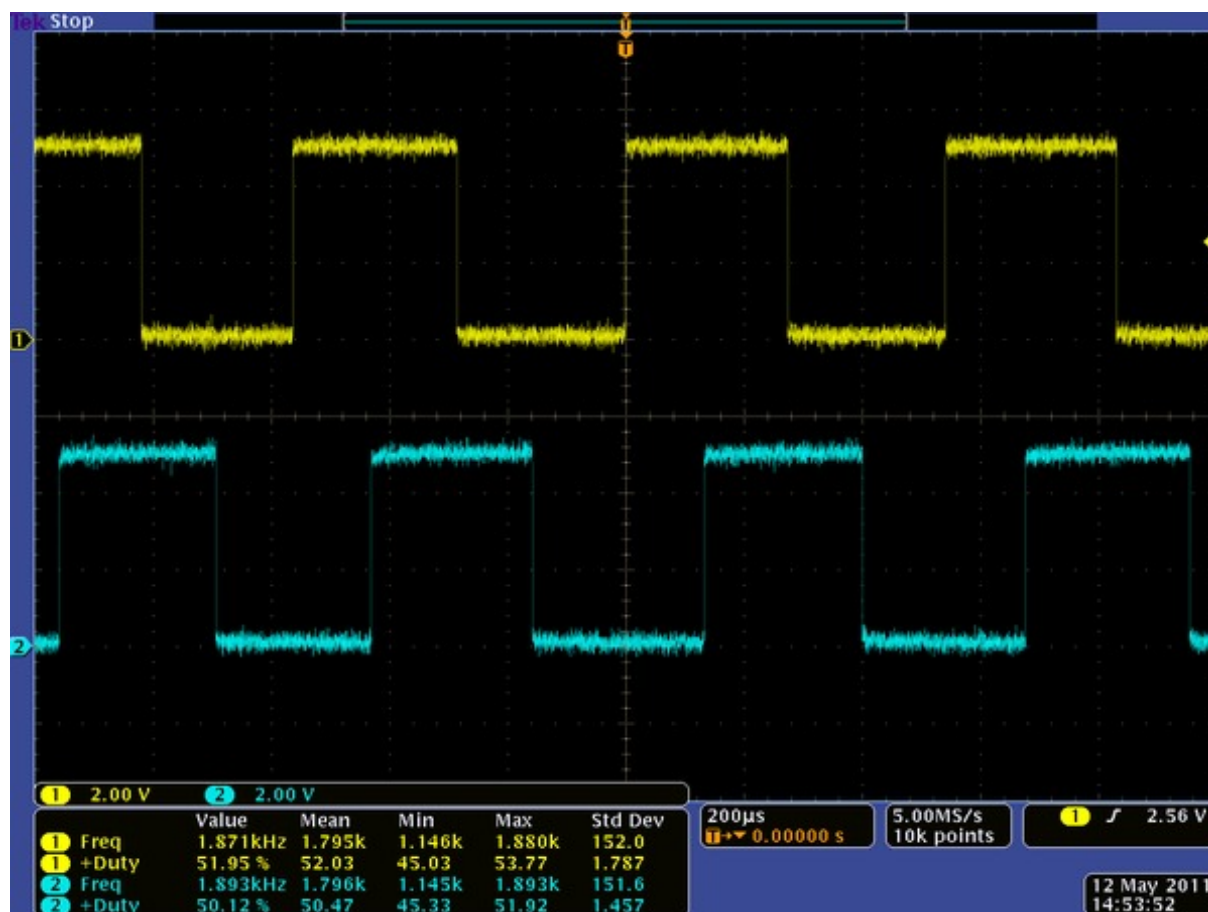
$$\frac{K_{ratio}}{JL_{ratio}^2 s^2 + (JR_{ratio}^2 + Ld)s + K^2_{ratio} + Rd}$$

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

$$\frac{J_{ratio}^2 s + d}{JL_{ratio}^2 s^2 + (JR_{ratio}^2 + Ld)s + K^2_{ratio} + Rd}$$

8.1.1 - 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 34:1, l'arbre moteur fait 34 tours lorsque l'arbre « principal » en fait 1
- Le codeur génère 48 impulsions à chaque fois qu'il fait un tour. Cependant, la carte Raspberry Pi présente des problèmes de performances pour gérer les interruptions générées par ces impulsions. Par conséquent, on ne prend en compte qu'une interruption sur 4
- 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 :

$$34 * 12 = 408 \text{ impulsions.}$$

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 2π radians) :

$$2\pi * N / (0.01 * 408)$$

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) :

$$2\pi / (T_s * CPR * \text{ratio})$$

avec :

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

Dans notre cas de figure, la résolution serait la suivante

$$2\pi / (0.01 * 408) = 1,54 \text{ rad/s}$$

Enfin, pour l'améliorer, nous faisons une moyenne sur les 40 derniers échantillons. Ceci introduit un retard (à prendre en compte dans les asservissements) mais permet d'avoir une résolution de l'ordre de 0.04 rad/s.

8.1.2 - Comptage du nombre d'impulsions

Dans le cas du robot X-Bot Raspberry Python, le comptage des impulsions revient à compter le nombre de fronts montants du signal jaune représenté sur l'image ci-dessus. Pour ce faire, la seule méthode viable consiste à brancher ce signal (le fil jaune du codeur utilisé) sur une des deux entrées « interruption » de la carte Pololu A-Star. Le fil blanc est branché sur une entrée digitale classique et les deux derniers fils (bleu et vert) seront respectivement branchés sur le 3.3 V et sur la masse de la carte de commande.

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. Pour connaître le sens de rotation du moteur, il faut donc mesurer le niveau de B lorsque l'interruption survient.

8.2 - Modèle du robot

La fonction de transfert du robot entre la tension de commande et la vitesse longitudinale est de la forme :

$$\frac{\text{vitesse}(s)}{\text{tension}(s)} = \frac{G}{as^2 + bs + c}$$

Avec :

$$G = K \text{ratio} \text{ rayon_roue}$$

$$a = JL \text{ratio}^2 + LM \text{rayon_roue}^2 + J_{\text{roue}} L$$

$$b = JR \text{ratio}^2 + L d_{\text{roue}} \text{rayon_roue}^2 + MR \text{rayon_roue}^2 + J_{\text{roue}} R + L d$$

$$c = K^2 \text{ratio}^2 + R d_{\text{roue}} \text{rayon_roue}^2 + R d$$

Les valeurs des paramètres sont les suivantes :

- rayon_roue : 4.5 cm
- M (masse du robot) : 400 g
- J_roue (inertie d'une roue) : 0.243e-4 kg.m²
- d_roue (amortissement dû au contact entre la roue et le sol) : dépend très fortement de la nature du sol (dur, moquette,...) , estimé à 0.1 N.s/m sur sol dur
- autres paramètres : ce sont les paramètres du moteur, voir ci-dessus

L'asservissement de vitesse longitudinale est un régulateur proportionnel intégral dont les gains sont les suivants :

- Gain proportionnel : 1
- Gain intégral : 10

8.3 - Commande des moteurs en tension (en Python)

Cette expérience permet de changer la vitesse de rotation des moteurs 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 des moteurs consiste à envoyer des signaux PWM sur la carte de commande moteur connectée à la carte Raspberry Pi afin de faire varier la tension d'alimentation des moteurs. 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-X-Bot.html>

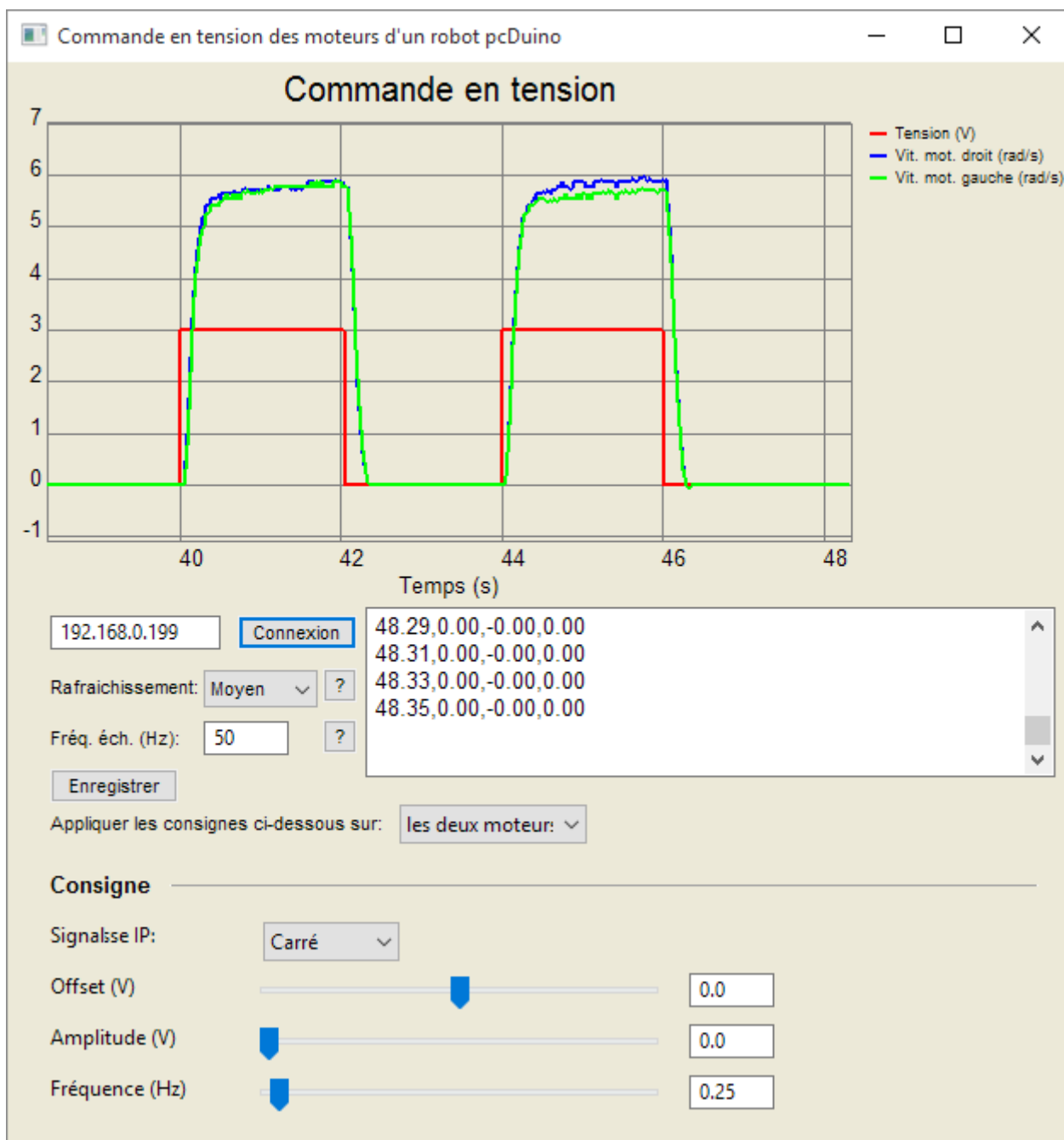
Son nom est de la forme `InterfaceReseauCommandeDeuxMoteursEnTension_x.y.zip` (x.y correspond au numéro de version du programme).

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

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

- mettre le système sous tension
- se connecter à la carte Raspberry Pi en Wifi ou en Ethernet sur votre réseau local. Dans le premier cas, l'adresse IP du robot est statique, fixée à 192.168.0.199. Dans le second cas, l'adresse IP du robot est fournie par le serveur DHCP de votre réseau local
- lancer l'application « `InterfaceReseauCommandeDeuxMoteursEnTension` »
- entrer l'adresse IP du robot dans la zone correspondante et cliquer sur « Connexion ». L'application démarre alors à distance le programme Python de commande des moteurs sur la carte Raspberry Pi. 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)
 - les vitesses mesurées (rad/s, en bleu et vert)
- zone de sélection de l'adresse IP: entrer l'adresse IP de votre robot 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)
- « Appliquer les consignes ci-dessous sur: »: permet d'appliquer les consignes sur le moteur gauche, le moteur droit ou les deux moteurs
- Zone « Consigne »: elle permet de modifier la consigne de tension avec les curseurs. La vitesse des moteurs 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 des deux moteurs, vous obtiendrez probablement une valeur différente, bien que la tension de commande soit 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.

8.4 - Asservissement des moteurs en vitesse (en Python)

Cette expérience permet d'asservir la vitesse de rotation des moteurs 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 des moteurs consiste à calculer, grâce à un régulateur de type PID, la tension de commande à appliquer aux moteurs (via commande PWM de la carte de commande moteur connectée sur la carte Raspberry Pi) 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:

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

avec :

- T_s : cadence d'échantillonnage (0.01 s)
- CPR : nombre d'impulsions par tour du codeur (48)
- ratio : rapport de réduction du moteur (34)

Ceci donne une résolution de $2\pi / (0.01 * 1632) = 0.4$ 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-X-Bot.html>

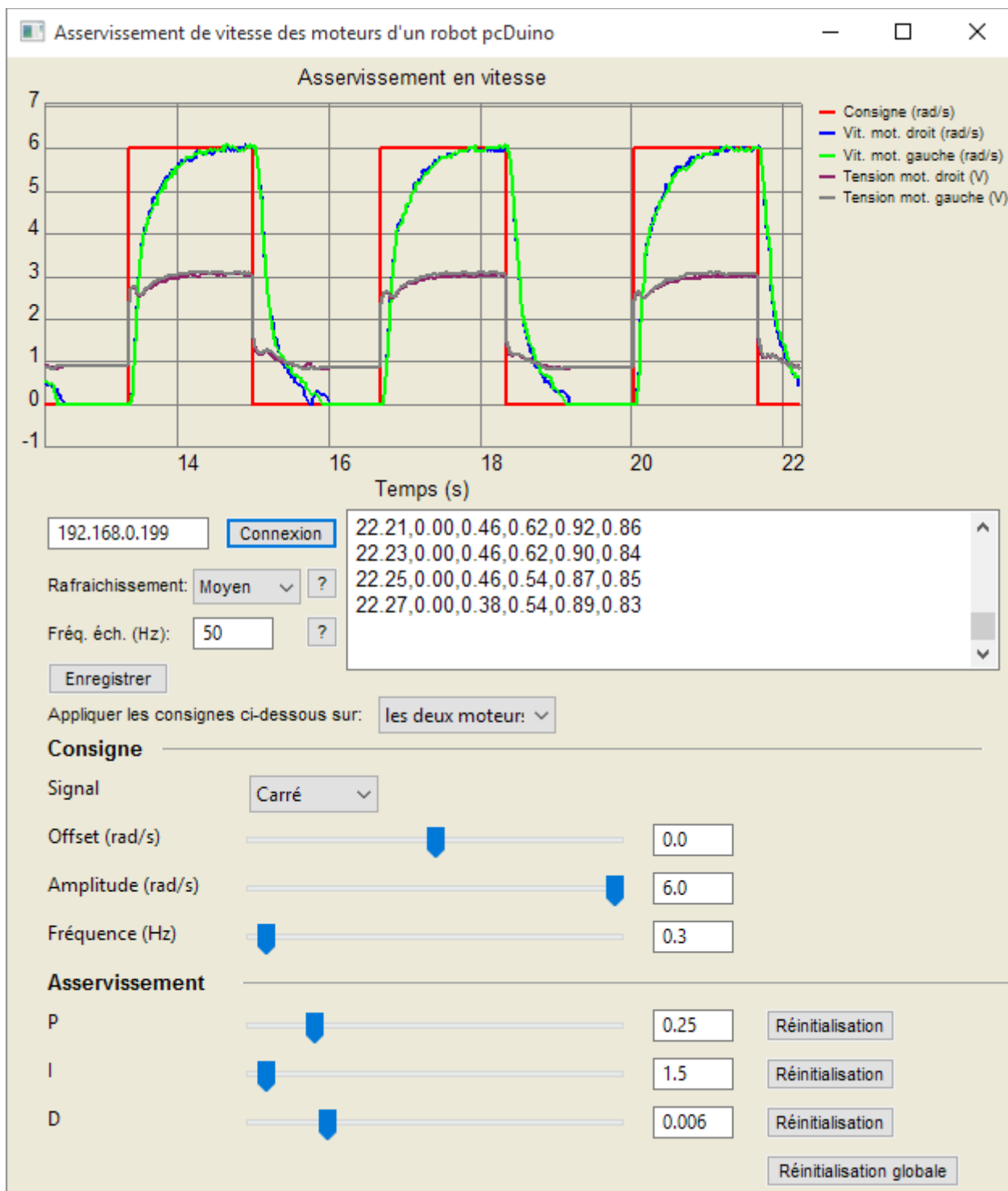
Son nom est de la forme InterfaceReseauAsservissementDeuxMoteursEnVitesse_x.y.zip (x.y correspond au numéro de version du programme).

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

- mettre le système sous tension
- se connecter à la carte Raspberry Pi en Wifi ou en Ethernet sur votre réseau local. Dans le premier cas, l'adresse IP du robot est statique, fixée à 192.168.0.199. Dans le second cas, l'adresse IP du robot est fournie par le serveur DHCP de votre réseau local
- lancer l'application «InterfaceReseauAsservissementDeuxMoteursEnVitesse»
- entrer l'adresse IP du robot dans la zone correspondante et cliquer sur « Connexion ». L'application démarre alors à distance le programme Python d'asservissement des moteurs sur la carte Raspberry Pi. 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)
 - les vitesses mesurées (rad/s, en bleu et en vert)
 - les tensions de commande (V, en marron et en gris)
- zone de sélection de l'adresse IP: entrer l'adresse IP de votre robot 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)
- « Appliquer les consignes ci-dessous sur: »: permet d'appliquer les consignes sur le moteur gauche, le moteur droit ou les deux moteurs
- Zone « Consigne »: elle permet de modifier la consigne de vitesse avec les curseurs. La vitesse des moteurs 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.

8.5 - Commande des moteurs en tension (en C/C++)

Cette expérience permet de changer la vitesse de rotation des moteurs en appliquant une tension variable par l'intermédiaire d'une application qui s'exécute sur votre ordinateur. La connexion entre l'ordinateur et le système A-Star (qui exécute le programme) se fait via une liaison série.

Le programme Arduino peut être téléchargé à l'adresse suivante:

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

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

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

Le principe de pilotage des moteurs consiste à envoyer des signaux PWM sur le pont en H intégré à la carte A-Star afin de faire varier la tension d'alimentation des moteurs. 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-X_Bot.html (voir dans la partie « Version Romeo »)

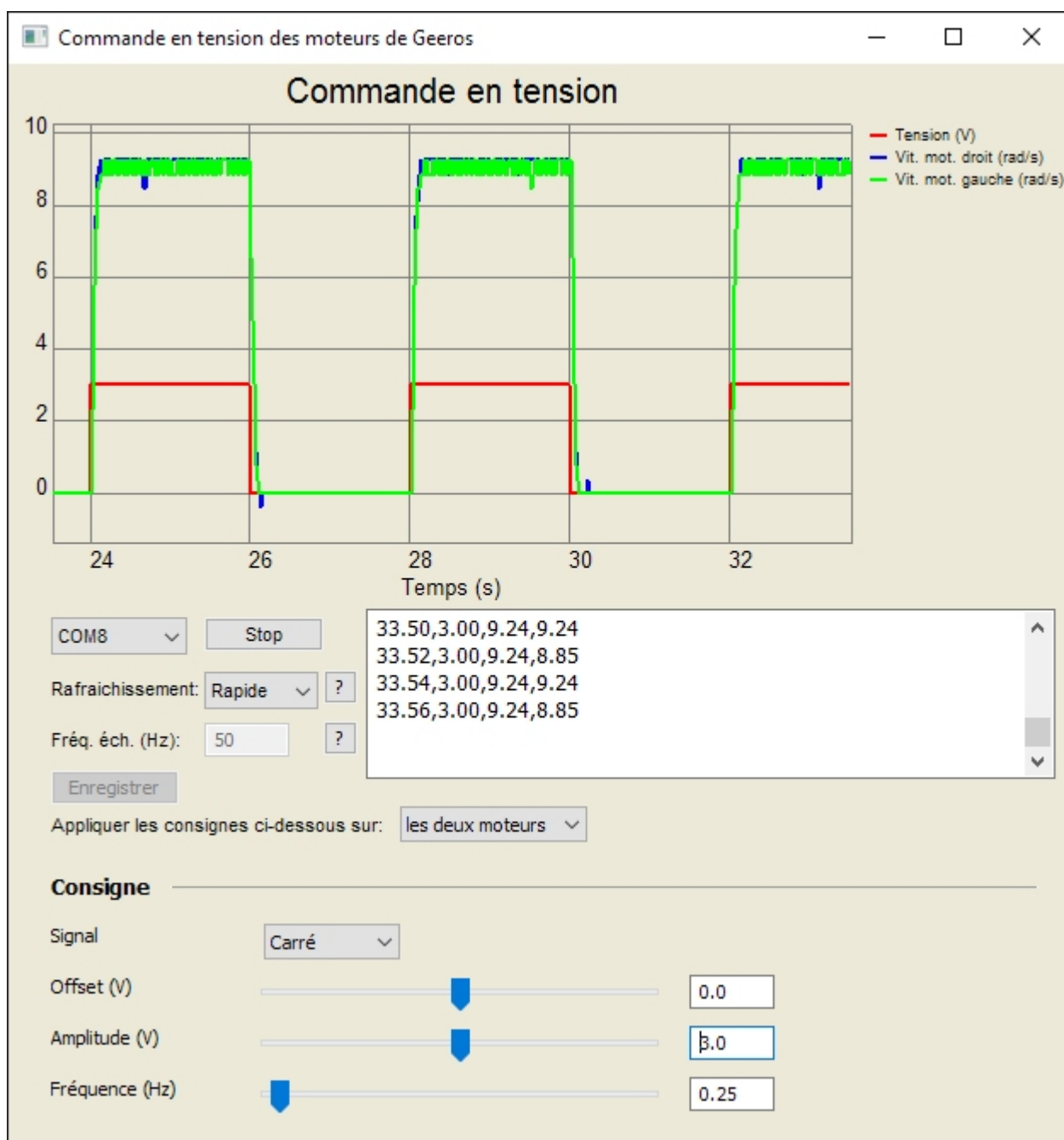
Son nom est de la forme GeerosInterfaceCommandeMoteursEnTension_x.y.zip (x.y correspond au numéro de version du programme).

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

Pour piloter la tension d'alimentation (et donc la vitesse) des moteurs depuis votre ordinateur, il vous suffit de suivre les étapes suivantes :

- mettre le système sous tension en positionnant le bouton marche-arrêt sur « I »
- connecter la carte A-Star à l'ordinateur avec le câble USB
- télécharger le programme Arduino « XBotAStarCommandeMoteursEnTension » sur la carte A-Star
- lancer l'application « GeerosInterfaceCommandeMoteursEnTension »

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)
 - les vitesses mesurées (rad/s, en bleu et vert)
- zone de sélection du port série: choisir le port série sur lequel est connectée votre carte A-Star 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 Arduino. Attention: ce paramètre doit être spécifié en Hz, alors que les valeurs correspondantes dans le programme Arduino (CADENCE_MS et TSDATA) sont spécifiées en ms. La relation entre les deux est la suivante:
$$\text{freq (Hz)} = 1000 / \text{cadence (ms)}$$
- 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)
- « Appliquer les consignes ci-dessous sur: »: permet d'appliquer les consignes sur le moteur gauche, le moteur droit ou les deux moteurs
- Zone « Consigne »: elle permet de modifier la consigne de tension avec les curseurs. La vitesse des moteurs 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 l'affichage ne suit pas les consignes même avec une fréquence de rafraîchissement très lente, veuillez diminuer la cadence d'échantillonnage dans le programme Arduino (augmenter les valeurs TSDATA et CADENCE_MS)

IMPORTANT !

Si vous comparez la vitesse de rotation des deux moteurs, vous obtiendrez probablement une valeur différente, bien que la tension de commande soit 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.

8.6 - Asservissement des moteurs en vitesse (en C/C++)

Cette expérience permet d'asservir la vitesse de rotation des moteurs en appliquant une consigne de vitesse par l'intermédiaire d'une application qui s'exécute sur votre ordinateur. La connexion entre l'ordinateur et le système A-Star (qui exécute le programme) se fait via une liaison série.

Le programme Arduino peut être téléchargé à l'adresse suivante:

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

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

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

Le principe de pilotage des moteurs consiste à calculer, grâce à un régulateur de type PID, la tension de commande à appliquer aux moteurs (via commande PWM du pont en H intégré sur la carte A-Star) pour qu'il suive la consigne de vitesse spécifiée.

Notez que dans ce programme, la vitesse mesurée est en fait la moyenne glissante des 10 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:

$2\pi / (Ts \cdot CPR \cdot \text{ratio})$

avec :

- Ts : cadence d'échantillonnage (0.01 s)
- CPR : nombre d'impulsions par tour du codeur (48)
- ratio : rapport de réduction du moteur (34)

Ceci donne une résolution de $2\pi / (0.01 \cdot 1632) = 0.4 \text{ rad/s}$. 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-X_Bot.html (voir dans la partie « Version Romeo »)

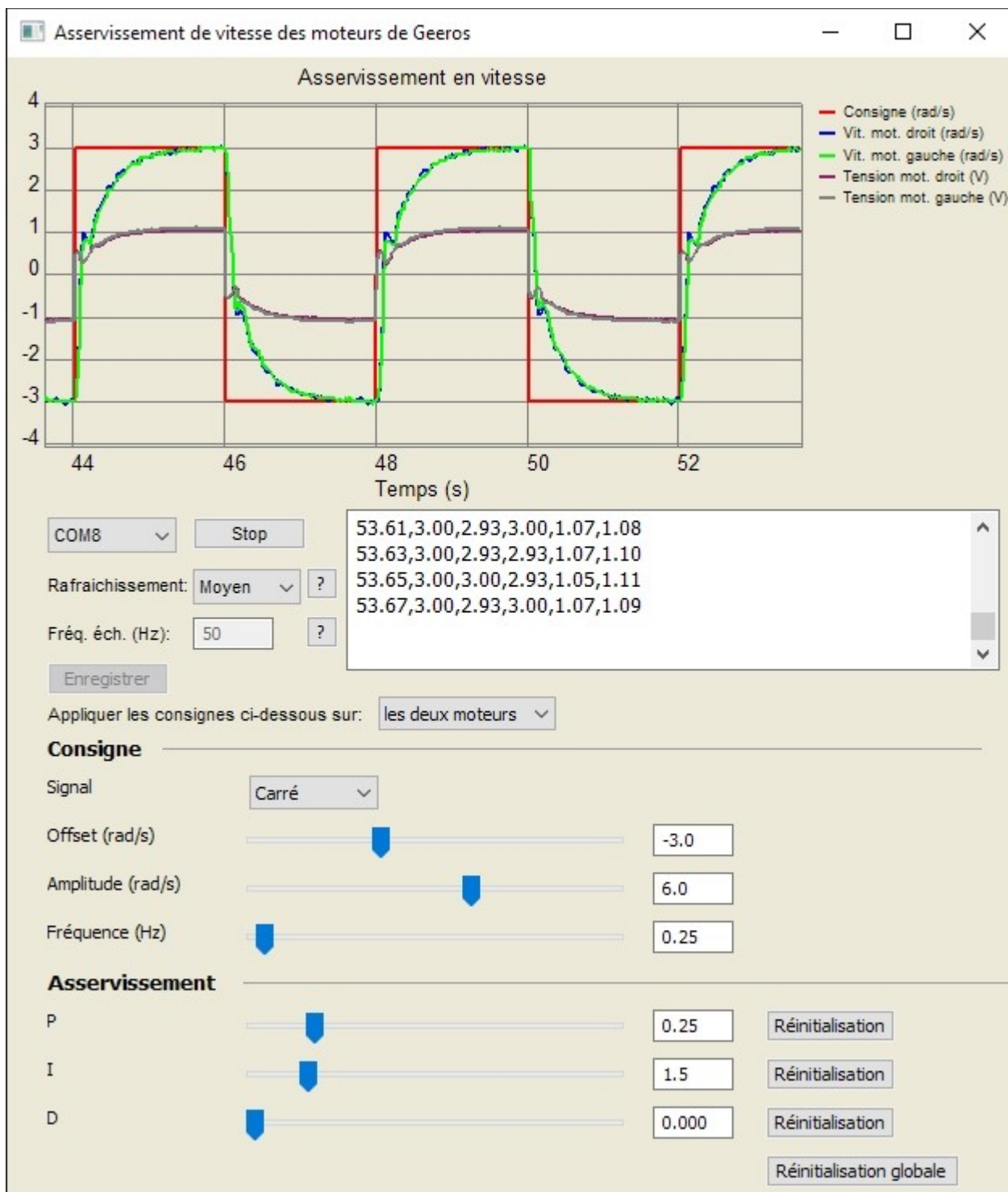
Son nom est de la forme GeerosInterfaceAsservissementMoteursEnVitesse_x.y.zip (x.y correspond au numéro de version du programme).

Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur GeerosInterfaceAsservissementMoteursEnVitesse_x.y.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):

- mettre le système sous tension en positionnant le bouton marche-arrêt sur « I »
- connecter la carte A-Star à l'ordinateur avec le câble USB
- télécharger le programme Arduino « XBotAStarAsservissementMoteursEnVitesse » sur la carte A-Star
- lancer l'application « GeerosInterfaceAsservissementMoteursEnVitesse »

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)
 - les vitesses mesurées (rad/s, en bleu et en vert)
 - les tensions de commande (V, en marron et en gris)
- zone de sélection du port série: choisir le port série sur lequel est connectée votre carte A-Star 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
- « Rafraichissement »: 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 Arduino. Attention: ce paramètre doit être spécifié en Hz, alors que la valeur correspondante dans le programme Arduino (TSDATA) est spécifiée en ms. La relation entre les deux est la suivante:
$$\text{freq (Hz)} = 1000 / \text{cadence (ms)}$$
- 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)
- « Appliquer les consignes ci-dessous sur: »: permet d'appliquer les consignes sur le moteur gauche, le moteur droit ou les deux moteurs
- Zone « Consigne »: elle permet de modifier la consigne de vitesse avec les curseurs. La vitesse des moteurs 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.

IMPORTANT !

Si l'affichage ne suit pas les consignes même avec une fréquence de rafraîchissement très lente, veuillez diminuer la cadence d'échantillonnage dans le programme Arduino (augmenter la valeur TSDATA). Ne pas modifier la valeur CADENCE_MS, sous peine de déstabiliser l'asservissement.

9 - Autre documentation disponible

Cette documentation, bien que baptisée « complète », ne détaille pas certains points qui sont présents dans une autre documentation disponible sur simple demande à l'adresse support@3sigma.fr :

- Documentation « développeur »: elle regroupe toutes les informations utiles pour ceux qui souhaitent entrer dans le cœur du système Linux de la carte Raspberry Pi. Cette documentation n'est pas en accès libre car elle contient des informations qui, utilisées à mauvais escient, peuvent corrompre le bon fonctionnement de votre X-Bot.

10 - Important

X-Bot Raspberry Python 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 (support@3sigma.fr).

Ne restez jamais bloqué sans nous contacter !

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