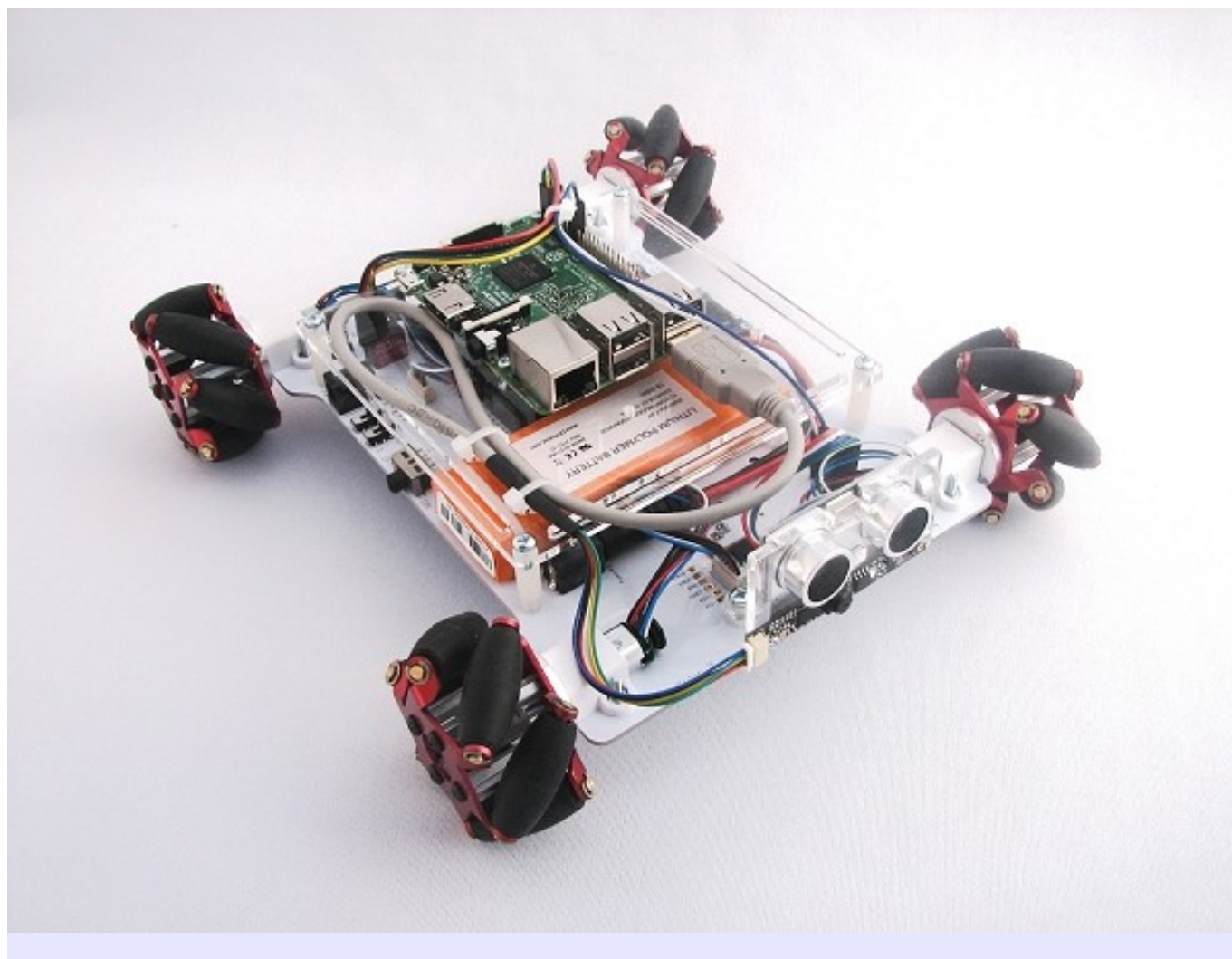




# T-QUAD

## IDENTIFICATION DES MOTEURS À COURANT CONTINU



Date de dernière mise à jour : 07/11/2017

## Table des matières

---

<b>1 - Introduction.....</b>	<b><u>3</u></b>
<b>2 - Essais à réaliser sur le système.....</b>	<b><u>4</u></b>
2.1 - Mesure du courant sur un échelon de tension à vitesse nulle.....	<u>5</u>
2.2 - Mesure de la vitesse de rotation sur un échelon de tension avec un couple résistant nul.....	<u>6</u>
<b>3 - Programme Arduino.....</b>	<b><u>8</u></b>
<b>4 - Application MyViz d'identification automatique.....</b>	<b><u>9</u></b>
<b>5 - Important.....</b>	<b><u>16</u></b>

## 1 - Introduction

Cette documentation présente la procédure à suivre pour l'identification des moteurs à courant continu du robot T-Quad et détaille la stratégie utilisée dans le programme Arduino permettant de réaliser cette opération.

Les programmes et applications associés à cette activité sont téléchargeables ici :

<https://github.com/3sigma/T-Quad-Identification-des-Moteurs-Courant-Continu>

## 2 - Essais à réaliser sur le système

Rappelons ci-dessous les équations d'un moteur à courant continu :

$$J_m n^2 \frac{d}{dt} \omega_m(t) + d \omega_m(t) - n K i_m(t) = Tr(t) \quad (2.1)$$

$$L \frac{d}{dt} i_m(t) + R_m i_m(t) = V(t) - K n \omega_m(t) \quad (2.2)$$

Avec :

- $R_m$  : résistance électrique interne
- $L$  : inductance des enroulements
- $J_m$  : moment d'inertie du rotor
- $K$  : constante de couple = constante de fem
- $d$  : coefficient de frottement visqueux
- $\omega_m$  : vitesse de rotation de l'arbre de sortie du réducteur
- $i_m$  : courant dans le moteur
- $V$  : tension d'alimentation
- $Tr$  : couple résistant
- $n$  : rapport de réduction (100)

On est donc en présence de deux équations électro-mécaniques couplées.

Afin d'identifier les paramètres du moteur, nous effectuerons les deux essais suivants :

- Mesure du courant sur un échelon de tension à vitesse nulle
- Mesure de la vitesse de rotation sur un échelon de tension avec un couple résistant nul

## 2.1 - Mesure du courant sur un échelon de tension à vitesse nulle

Lorsque la vitesse est nulle, l'équation (2.2) s'écrit :

$$L \frac{d}{dt} i_m(t) + R_m i_m(t) = V(t) \quad (2.3)$$

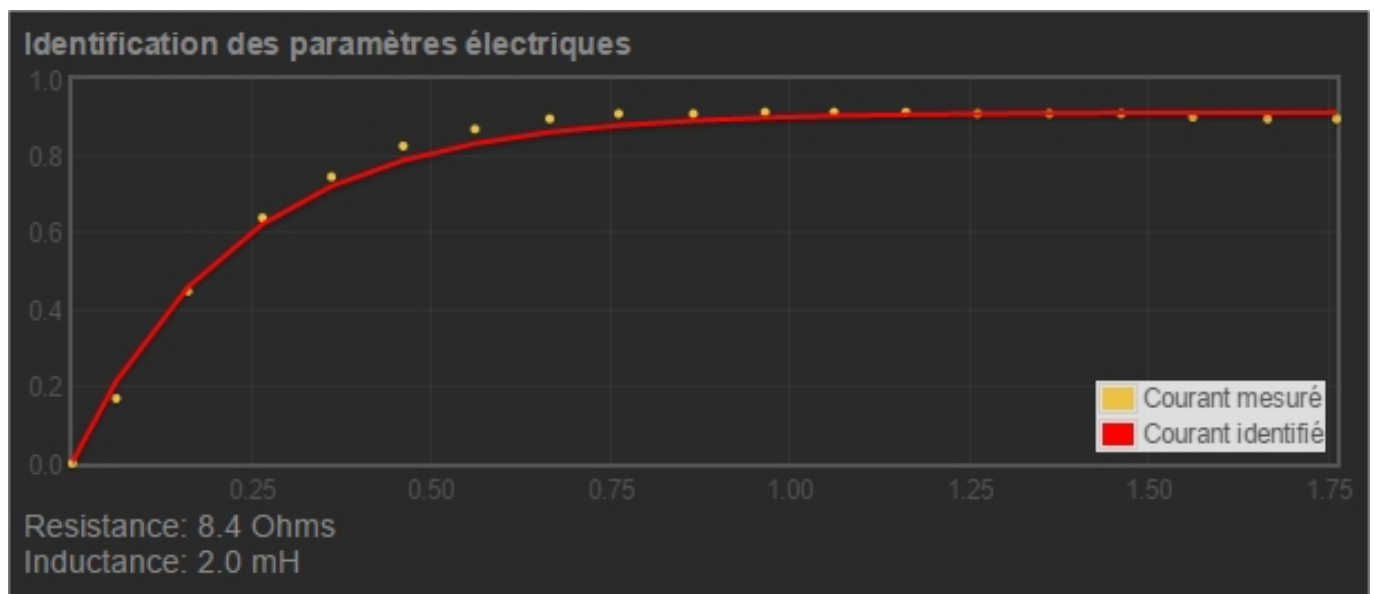
On peut donc facilement estimer l'inductance et la résistance à partir de la mesure du courant, puisqu'on est en présence d'un système du premier ordre.

Ce type d'essai se réalise traditionnellement en bloquant le rotor du moteur, ce qui présente deux inconvénients :

- ce n'est pas toujours très facile à réaliser d'un point de vue mécanique
- cela peut conduire à une surchauffe, voir à une destruction de certains composants si l'essai dure trop longtemps

La stratégie adoptée sur le robot et mise en œuvre dans le programme Arduino d'identification des moteurs consiste à capturer le courant sur un échelon initial très court avec la tension maximale délivrée par la batterie, alors que le moteur n'a pas encore commencé à tourner. Afin de capturer cette très courte période avec la meilleure résolution possible, le temps de conversion des convertisseurs analogiques-numériques de l'Arduino est accélérée d'un facteur 16 dans ce programme, pour qu'on puisse faire une mesure toutes les 100  $\mu$ s.

L'application d'identification qui s'exécute avec le logiciel MyViz (voir plus loin) permet ainsi d'obtenir ce type de capture :



Noter que l'unité de l'abscisse est la milliseconde.

## 2.2 - Mesure de la vitesse de rotation sur un échelon de tension avec un couple résistant nul

La mise en vitesse du moteur se fait avec une constante de temps beaucoup plus importante que la montée en courant que nous venons de voir. Pour faciliter les choses, nous allons par conséquent négliger l'inductance dans ce qui suit.

Si on annule en plus le couple résistant (le moteur tournant donc à vide), les équations (2.1) et (2.2) se combinent en :

$$J_m n^2 \left( \frac{d}{dt} \omega_m(t) \right) R_m + (K^2 n^2 + R_m d) \omega_m(t) = K V(t) n \quad (2.4)$$

Nous retrouvons, comme précédemment, un premier ordre. Cependant, nous devons identifier ici 3 paramètres :  $J_m$ ,  $K$  et  $d$ . Il faut donc ajouter une équation car la courbe de réponse d'un premier ordre ne permet d'identifier que 2 paramètres.

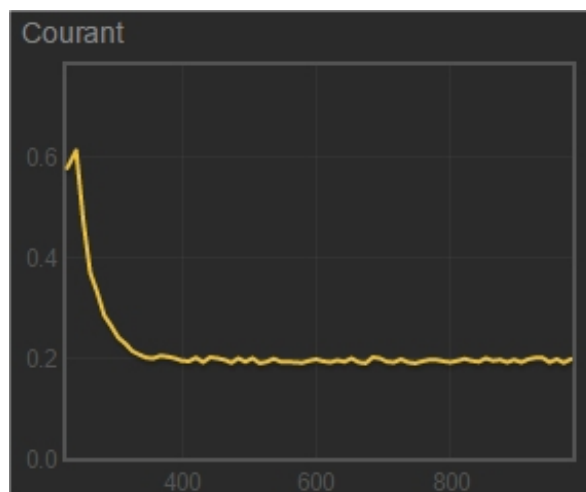
Cette équation est celle de la vitesse en régime permanent (lorsque l'accélération est nulle) issue de (2.1).

$$d\omega_m(t) - nK i_m(t) = 0 \quad (2.5)$$

La mesure de la vitesse et du courant en régime permanent nous permettra donc de remonter à la relation entre la constante de couple et le coefficient de frottement.

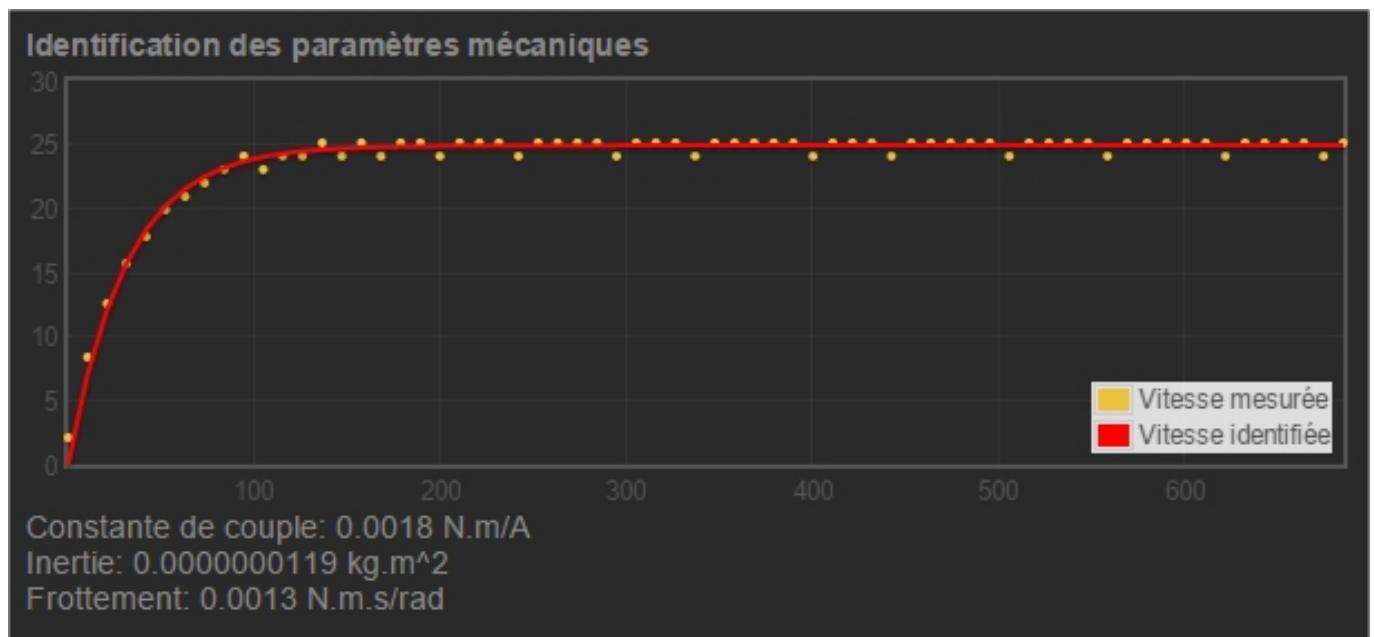
La stratégie adoptée sur le robot et mise en œuvre dans le programme Arduino d'identification des moteurs consiste à calculer à la fin de chaque période de PWM une moyenne des courants mesurés pendant cette période écoulée. La fréquence des PWM étant de 500 Hz, ceci permet d'avoir une mesure toutes les 2 ms, réalisée sur la moyenne de 20 échantillons de courant (la cadence des mesures de courant étant de 100  $\mu$ s).

Le profil de courant obtenu dans MyViz lors de l'essai en vitesse est le suivant :



Seule la valeur en régime permanent nous est utile, mais il est cependant intéressant de constater que cette méthode permet d'obtenir un tracé de très bonne qualité dans la phase transitoire

La vitesse est quant à elle mesurée d'une façon plus classique, à partir du comptage des interruptions générées par le codeur incrémental sur des intervalles de temps de 10 ms. L'identification se fait ensuite facilement à partir de cette mesure de vitesse :



Noter que contrairement à la stratégie décrite au paragraphe 2.1, la tension d'alimentation des moteurs est fixée ici à 6 V.

### 3 - Programme Arduino

Le programme Arduino mettant en œuvre la stratégie présentée plus haut est disponible à l'adresse suivante, en fonction de la version de matériel (écrite juste à côté de l'écran OLED) :

- Version 1.1  
[https://raw.githubusercontent.com/3sigma/T-Quad-Identification-des-Moteurs-Courant-Continu/master/Arduino/Identification/Identification\\_1.1.ino](https://raw.githubusercontent.com/3sigma/T-Quad-Identification-des-Moteurs-Courant-Continu/master/Arduino/Identification/Identification_1.1.ino)
- Version 1.2 et supérieur  
<https://raw.githubusercontent.com/3sigma/T-Quad-Identification-des-Moteurs-Courant-Continu/master/Arduino/Identification/Identification.ino>

Il se trouve également dans l'archive suivante :

<https://github.com/3sigma/T-Quad-Identification-des-Moteurs-Courant-Continu/archive/master.zip>

Il est nécessaire de télécharger ce programme sur la carte Arduino Mega pour réaliser cette procédure d'identification.

La procédure à suivre pour la programmation est la suivante:

- Si ce n'est pas déjà fait, installez l'IDE Arduino et les bibliothèques additionnelles nécessaires (voir la documentation générale : [https://github.com/3sigma/T-Quad/raw/master/T-Quad\\_General.pdf](https://github.com/3sigma/T-Quad/raw/master/T-Quad_General.pdf))
- Ouvrez le programme Identification.ino téléchargé précédemment dans l'IDE Arduino
- Si votre T-Quad possède une carte de communication (pcDuino, Raspberry Pi ou autre), débranchez le câble de communication à 4 fils situé juste à côté du moteur arrière droit
- Mettre le robot sous tension en appuyant sur le bouton marche-arrêt
- Connecter le câble USB reliant l'ordinateur et la carte Arduino Mega
- Lancer le téléchargement

L'essai, réalisé sur le moteur arrière gauche, démarre tout de suite à la fin du téléchargement.

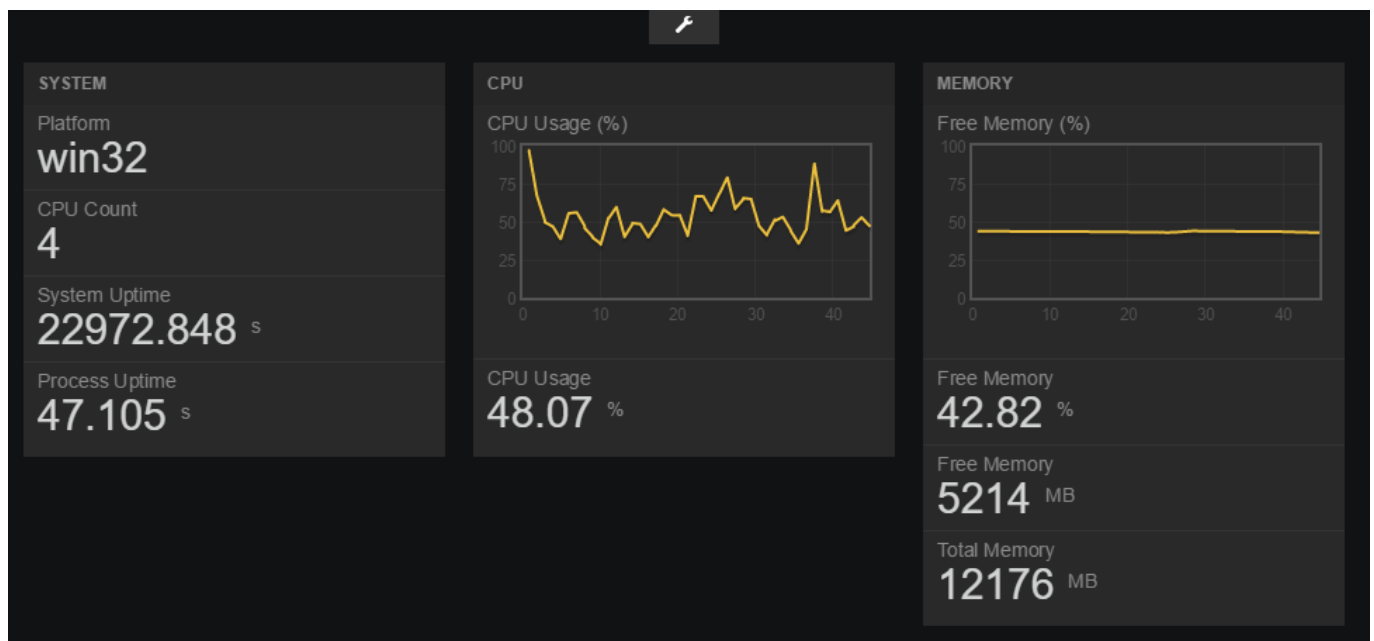
A la fin de l'essai, un message s'affiche sur l'écran OLED du robot, indiquant la tension d'alimentation du moteur lors de l'essai initial en courant. Cette valeur est nécessaire pour paramétrer l'application MyViz d'identification automatique des paramètres du moteur.



## 4 - Application MyViz d'identification automatique

Les activités réalisées sur le robot T-Quad utilisent en général le logiciel MyViz, très souple pour créer des tableaux de bord de pilotage et de visualisation de données.

Après l'avoir téléchargé (<http://www.3sigma.fr/Telechargements-MyViz.html>) et installé, lancez son exécution. Le tableau de bord initialement affiché sera similaire à la capture d'écran ci-dessous :



Ce tableau de bord n'est qu'un exemple de ce qui peut être réalisé avec MyViz. Nous verrons plus loin comment charger celui correspondant à l'expérience que nous souhaitons réaliser dans ce chapitre.

Pour réaliser la procédure d'identification automatique, les conditions suivantes doivent être remplies :

- si votre T-Quad possède une carte de communication (pcDuino, Raspberry Pi ou autre), le câble de communication à 4 fils situé juste à côté du moteur arrière droit doit être débranché
- le robot doit être allumé
- le câble USB doit relier l'ordinateur et la carte Arduino Mega
- le programme Arduino d'identification doit être chargé (voir chapitre 3)

Charger ensuite le tableau de bord d'identification automatique dans MyViz. Pour cela, il faut tout d'abord récupérer ce dernier sur votre ordinateur, à partir du lien suivant :

[https://raw.githubusercontent.com/3sigma/T-Quad-Identification-des-Moteurs-Courant-Continu/master/MyViz/T-Quad\\_Identification.json](https://raw.githubusercontent.com/3sigma/T-Quad-Identification-des-Moteurs-Courant-Continu/master/MyViz/T-Quad_Identification.json)

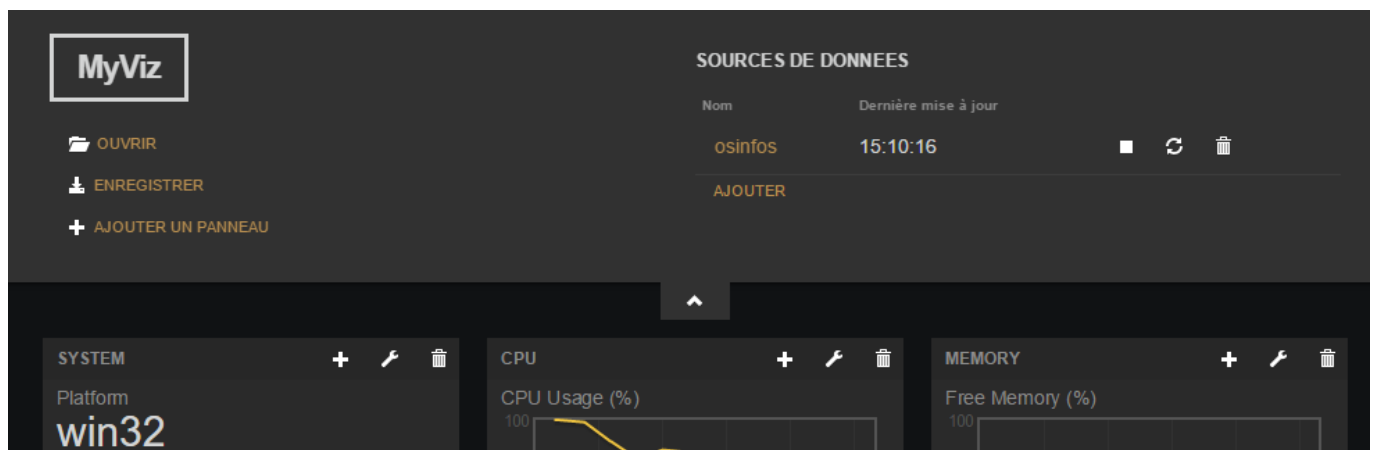
Il se trouve également dans l'archive suivante :

<https://github.com/3sigma/T-Quad-Identification-des-Moteurs-Courant-Continu/archive/master.zip>

Pour l'ouvrir dans MyViz, il suffit ensuite de cliquer sur la clé en haut de la fenêtre :

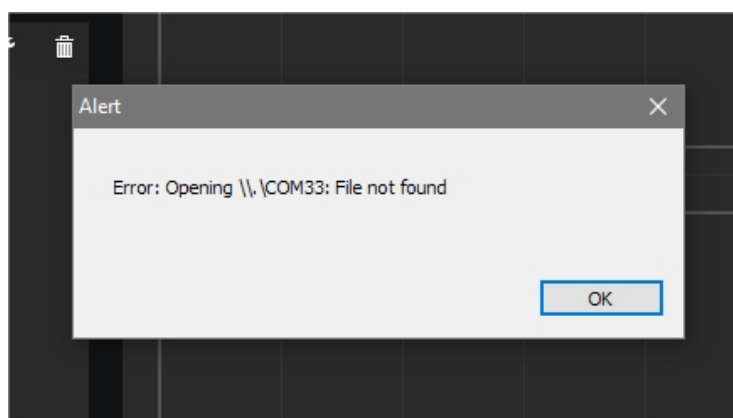


Ceci permet de déplier le panneau supérieur :



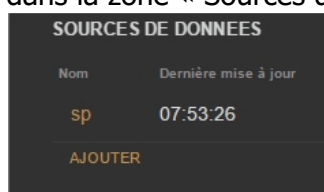
Cliquez sur « Ouvrir » et sélectionnez le fichier T-Quad\_Identification.json que vous venez de télécharger.

Sauf hasard extraordinaire, vous devriez voir apparaître le message d'erreur suivant :



Ce message est normal, il indique simplement que le port série sur lequel votre robot est branché n'est pas le même que celui enregistré dans le tableau de bord. Pour le changer, il suffit de suivre ces étapes :

- cliquer sur la clé pour déplier le panneau supérieur de MyViz
- dans la zone « Sources de données », cliquez sur « sp » :



- il s'ouvre alors la fenêtre suivante, permettant le paramétrage de la communication série :

SOURCE DE DONNÉES

Flux de données temps-réel provenant du port série.

TYPE

Port série

NOM

sp

PORT

COM33

NOMBRE DE BAUDS

115200

VARIABLES À LIRE

temps,courant,omega

Nom des variables à lire, séparées par des virgules

VARIABLES À ENVOYER

Nom des variables à envoyer, séparées par des virgules

TAUX DE RAfraîchissement POUR L'ENVOI DES DONNÉES

2000

MILLISECONDES

Refresh rate for sending data (> 1000 ms). Data will be sent even if control values are not changed

SÉPARATEUR

,

Séparateur des données reçues (et éventuellement envoyées)

MÉTHODE DE "SOMME DE CONTRÔLE"

Aucune

Si des données sont envoyées, une "somme de contrôle" sera automatiquement ajoutée dans une variable "\_cro", calculée à partir des autres valeurs

ENREGISTRER

ANNULER

- dans la zone « Port », indiquez le port série sur lequel votre carte Arduino Mega est connectée, puis cliquez sur le bouton « Enregistrer »
- vous allez entendre le moteur arrière gauche tourner brièvement, voir s'afficher dans MyViz un premier résultat d'identification et sur l'écran OLED un message du type :  
«  
Tension d'alim  
pour essai  
en courant :  
7.40 V  
»

Ce message indique la tension avec laquelle l'essai en courant a été fait. Rappelons que lors de cet essai, on alimente le moteur à la tension maximale pendant un très bref instant pour qu'il n'ait pas le temps de tourner. Il faut maintenant terminer le paramétrage du tableau de bord avec cette tension pour que l'identification donne des résultats corrects.

- cliquer sur la clé en haut à droite de la courbe d'identification en courant :



Attention : vous devez cliquer sur la clé du bas, qui n'apparaît que lorsqu'on approche la souris dans cette zone. La clé du haut permet de paramétrer le panneau complet alors que cette clé du bas permet de paramétrer le « widget » d'identification contenu dans ce panneau

- la fenêtre de paramétrage apparaît alors :

**WIDGET**

Identification des paramètres d'un moteur à courant continu.

TYPE	Identification	
TEMPS (AXE X)	<code>datasources["sp"]["temps"]</code>	+ DATASOURCE JS EDITOR
COURANT	<code>datasources["sp"]["courant"]</code>	+ DATASOURCE JS EDITOR
VITESSE DE ROTATION	<code>datasources["sp"]["omega"]</code>	+ DATASOURCE JS EDITOR
FENÊTRE EN COURANT (MS)	2.0	
	Longueur de la fenêtre d'identification en courant	
DÉBUT DU RÉGIME PERMANENT SUR LA VITESSE (MS)	500	
	Prenez une marge pour être sûr qu'après cet instant la vitesse a atteint le régime permanent	
TENSION MOTEUR ESSAI EN COURANT (V)	7.63	
TENSION MOTEUR ESSAI EN VITESSE (V)	6	
RAPPORT DE RÉDUCTION	100	

ENREGISTRER ANNULER

Dans la zone « Tension moteur essai en courant », indiquez la tension lue sur l'écran OLED. Dans la zone « Tension moteur essai en vitesse », laissez 6 V. Cliquez ensuite sur le bouton « Enregistrer »

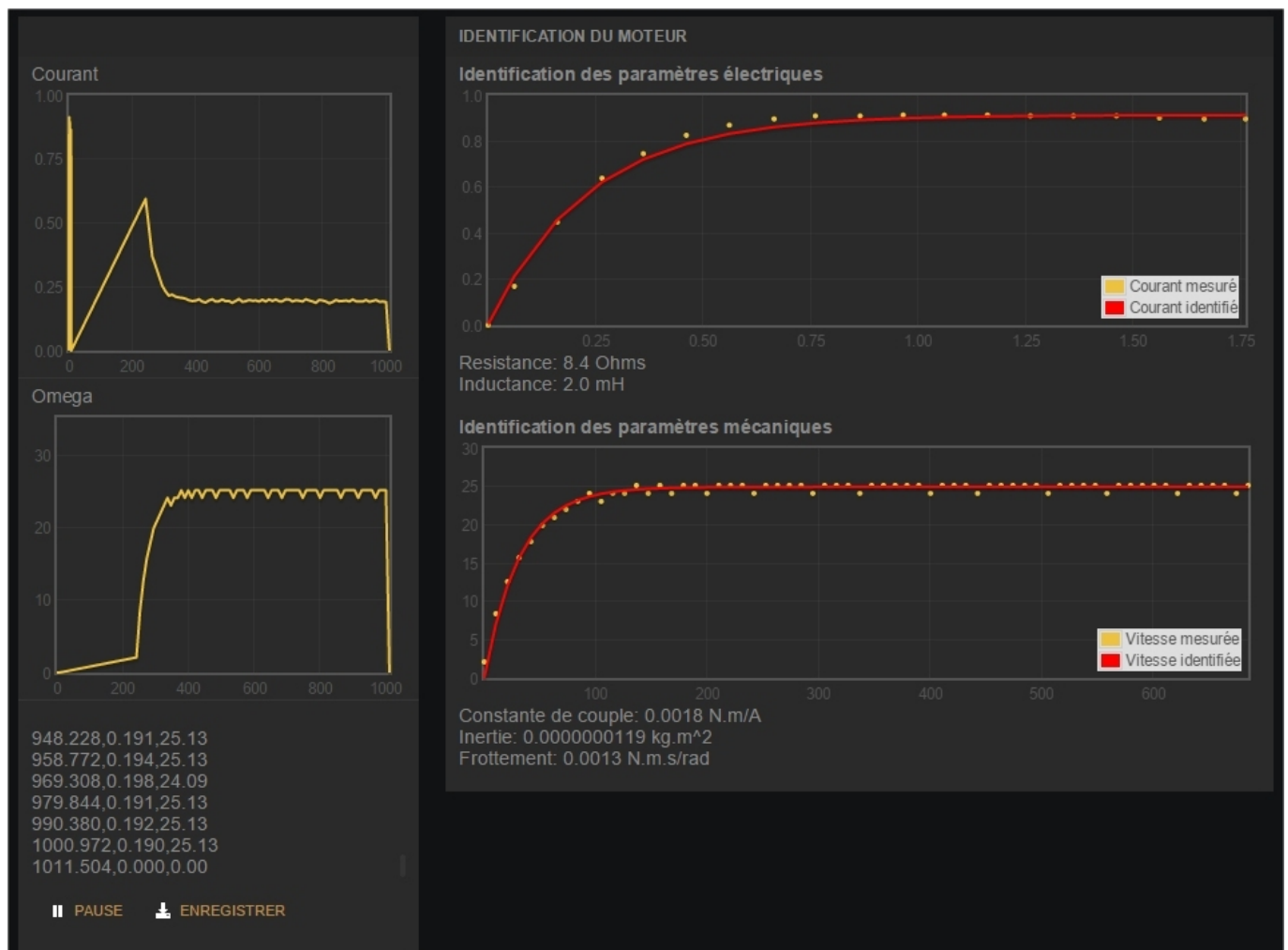
- Relancez l'identification : il faut pour cela couper la liaison série en cliquant sur le bouton « Stop » de la zone « Sources de données »



puis cliquer sur le bouton de rechargement à côté :



Vous obtenez alors un résultat d'identification correct :



Remarques :

- à condition d'avoir replié le panneau supérieur de MyViz, il est possible de zoomer avec la souris dans les tracés de la colonne de gauche. Pour dézoomer, double-cliquer dans la fenêtre
- vous pouvez recommencer la procédure d'identification autant que vous le souhaitez. Vous devez pour cela, à chaque fois, cliquer sur le bouton d'arrêt puis sur le bouton de rechargement de la liaison série (sp) dans la zone des sources de données
- si vous souhaitez charger un autre programme Arduino, il faut au préalable arrêter la communication série dans MyViz en cliquant sur le bouton d'arrêt

## 5 - Important

T-Quad est un produit « vivant » en constant développement pour l'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](mailto:support@3sigma.fr)).

**Ne restez jamais bloqué sans nous contacter !**

Pour tout problème ou toute requête, contactez-nous à l'adresse [support@3sigma.fr](mailto:support@3sigma.fr)