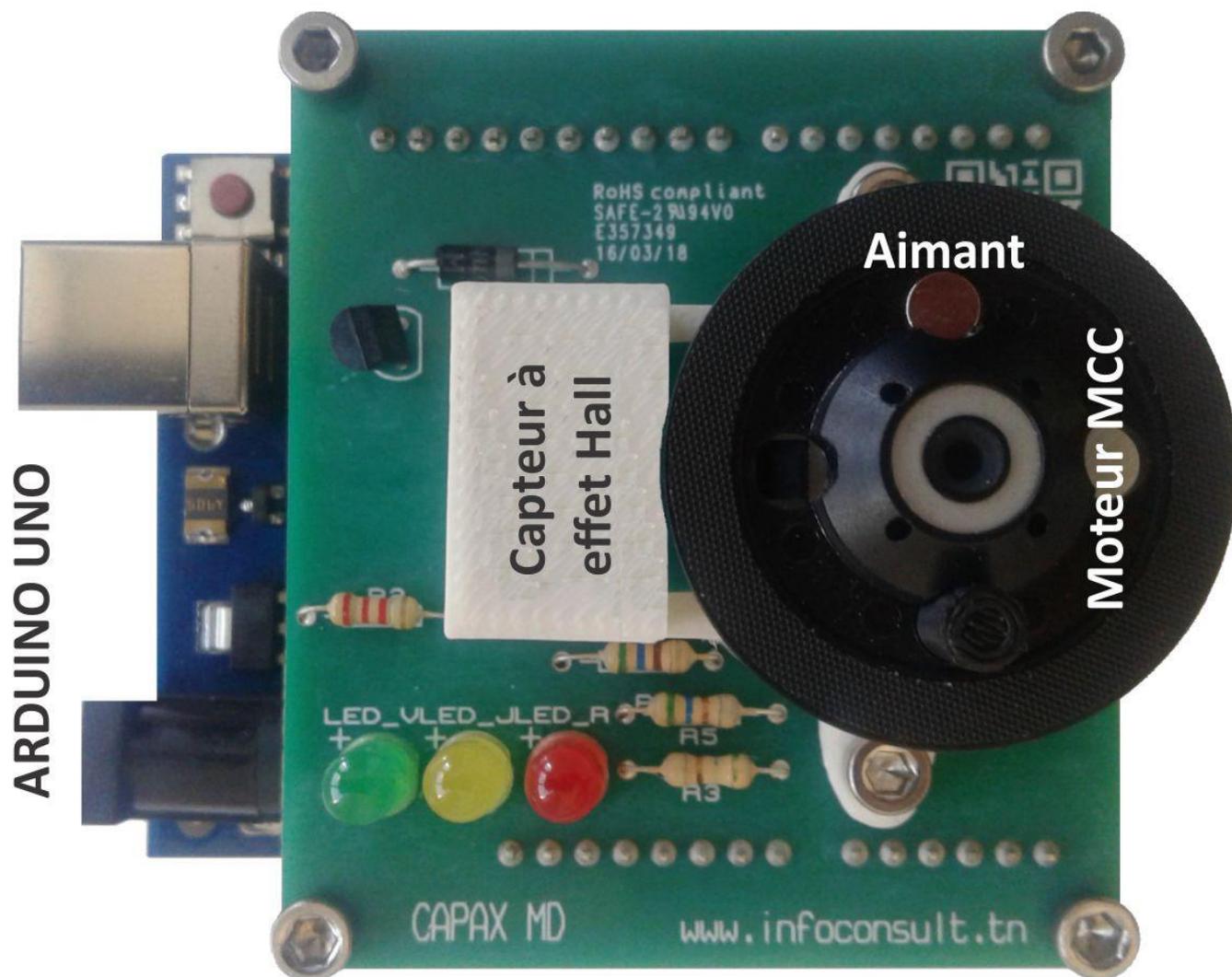


Identification du Moteur à Courant Continu à Aimant Permanent



SEQUENCE DE TRAVAUX PRATIQUES N°2

TABLE DES MATIERES

MISE EN SITUATION	3
1- Identification	3
2- Acquisition de données suite à une excitation.....	4
3- Identification de la fonction de transfert d'un MCC et étude temporelle	5
4- Etude fréquentielle.....	9
ANNEXES	10
Principe de la mesure de la vitesse de rotation	12
Principe de la Séquence binaire pseudo-aléatoire (SBPA)	13
Implémentation de l'identification	15
BIBLIOGRAPHIE	15

Objectif :

En abordant un problème de commande de point de vue formulation, il y a des anomalies spécifiques entre le système concret réel et le modèle mathématique mis au point dans le but de concevoir la loi de commande. Cette particularité peut être due à la variation des paramètres du système ou à l'approximation du comportement complexe du système par un modèle direct. Pour cela nous devons exciter le moteur par un signal riche en fréquence pour déterminer les caractéristiques de sa fonction de transfert en utilisant Matlab/Simulink.

MISE EN SITUATION

1- Identification

L'identification de système est une technique de l'automatique consistant à obtenir un modèle mathématique d'un système à partir de mesures. Dans la pratique, deux démarches sont possibles pour l'obtention du modèle d'un système. La première approche est analytique et la deuxième est expérimentale [1]. La modélisation analytique utilise les équations physico-chimiques qui régissent le comportement dynamique du système. La modélisation expérimentale traite le système comme une boîte noire et utilise les mesures expérimentales pour extraire le modèle. Bien que la modélisation analytique offre une description précise facilitant la compréhension du comportement des procédés, elle est souvent complexe et peu utilisée pour la commande. Les automaticiens utilisent surtout la modélisation expérimentale car elle présente une grande souplesse lors de la synthèse des régulateurs. Cette dernière modélisation comporte les modèles de type entrée-sortie, appelés aussi modèles de fichiers [1] [2] [3] [4].

Pour obtenir un modèle consistant, il est important d'exciter le processus avec toutes les fréquences de sa plage de fonctionnement. Le signal d'entrée appliqué doit donc être riche en fréquences (posséder un large spectre). En général on applique un signal périodique pseudo-aléatoire (PRBS). Lorsque le système possède plusieurs entrées/plusieurs sorties, il est important d'appliquer des signaux décorrélés pour ne pas introduire de biais d'identification. Une idée commune consistant à exciter l'une après l'autre les entrées est une mauvaise méthode car elle introduit un biais d'identification et ne rend pas compte du fonctionnement normal du système [5][6][7][8]. Il est important de respecter une procédure rigoureuse pour identifier un procédé :

- **Détermination du protocole de test:** de propriétés statistiques des signaux d'entrée pour balayer toutes les fréquences intéressantes.
- **Détermination de la structure du modèle :** type de modèle, ordre et retard.
- **Identification :** choix d'un algorithme pour trouver le modèle en minimisant les erreurs entre les mesures et le modèle, en général algorithme basé sur la méthode des moindres carrés (LS, RLS, RELS).
- **Validation du modèle :** Réalisation de plusieurs tests de vérification. Il est nécessaire pour cette étape d'utiliser des mesures différentes de celles utilisées lors de l'identification.

Dans cette manipulation, on s'intéresse par l'identification expérimentale du moteur, en excitant le système par une entrée riche en fréquence telle que la séquence binaire pseudo-aléatoire (**Annexe A2**). La méthode d'identification de Matlab «**System identification**» sera utilisée afin de déterminer le modèle du moteur MCC.

2- Acquisition de données suite à une excitation

Le moteur est excité en boucle ouverte par un échelon ou une Séquence Binaire Pseudo Aléatoire de rang 10, pour créer une réponse correspondant à la vitesse de rotation du moteur qui sera transmise à Matlab.

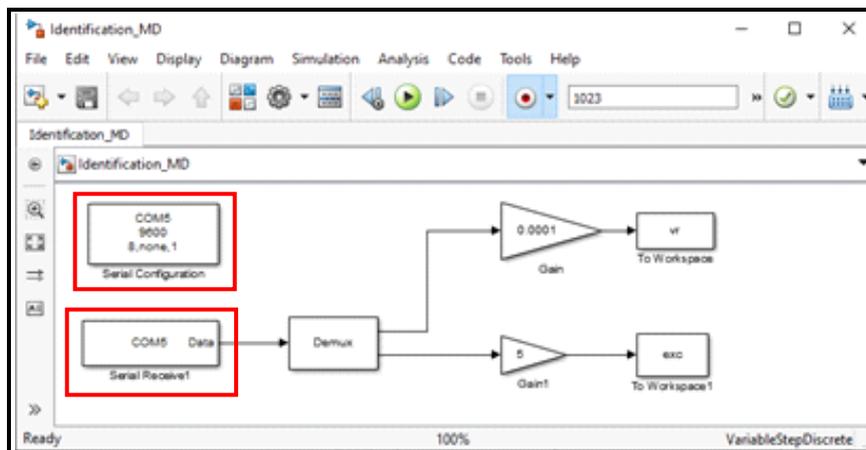


Travail à faire :

- 1- Brancher la carte **ARDUINO UNO**. La carte doit être préalablement installée. **Dans tous les cas, se référer au fichier [CAPAX-MD® Installation.pdf](http://www.infoconsult.tn/fr/telechargements.html) téléchargeable à partir de <http://www.infoconsult.tn/fr/telechargements.html>.**
- 2- Lancer l'environnement **ARDUINO**, ouvrir le fichier **TP2_MD_Identification.ino** puis vérifier le port COM.
- 3- Spécifier la période d'échantillonnage voulue et choisir le type de l'excitation (Echelon ou SBPA).
- 4- Mettre le CD-ROM "charge moteur" sur le plateau du moteur.
- 5- Téléverser le fichier **TP2_MD_Identification.ino** dans la carte **ARDUINO UNO**.
- 6- Lancer **Matlab** puis Lancer **Simulink**. **Attention : la carte ARDUINO UNO doit être d'abord connectée au PC avant de démarrer Matlab, sinon elle ne sera reconnue par ce dernier.**
- 7- Ouvrir le fichier **Identification_MD.slx** et y positionner correctement le port COM.

```

1 //*****
2 // TP2 (CAPAX-MD) : Identification MCC
3 // V1.0 Le 14/08/2018
4 // CAPAX SYSTEMS By www.infoconsult.tn
5 //*****
6
7 # define MCC_pin 9 // Commande de la vitesse du "MCC"
8 # define HALL_pin 2 // Lecture des impulsions du capteur à 60°
9 # define LED_V 3 // Led Vert
10 # define LED_O 5 // Led Orange
11 # define LED_R 6 // Led Rouge
12
13 //****Spécifier Te et le type de l'identification
14
15 int Te = 15000; // période d'échantillonnage [microseconde]
16 int Ident_type = 0; // 1 : Echelon, 0: SBPA
17
18 //****Spécifier Te et le type de l'identification
19
20
21
22 volatile unsigned long time; // Déclaration de la variable "t"
23 volatile unsigned long AncienTemps = 0; // Déclaration de la variable "t"
24 volatile unsigned long Duree = 0; // Déclaration de la variable "t"
  
```



- 8- **Le CD-ROM doit être à l'arrêt**, cliquer sur le bouton pour démarrer l'opération d'identification. La LED Jaune s'allume pendant 5 secondes.
- 9- La LED rouge s'allume, le CD-ROM commence à tourner indiquant le début de l'excitation du moteur. 1023 échantillons sont ainsi recueillis et transmis à l'environnement **Matlab**. La fin de l'excitation du moteur par le signal est indiquée par la LED verte. **Matlab** continue à recevoir les données jusqu'à la fin de la transmission.

10- Au niveau de **Workspace de Matlab**, il y a apparition de 2 tableaux : **exc** (valeurs de l'excitation) et **vr** (réponse du système à l'excitation). Contrôler la valeur minimale et maximale de **vr**. L'expérience est valide si **vr_min = 0** et **vr_max ≠ 0**.

Name	Value	Min	Max
exc	<1024x1 double>	0	5
tout	<1000x1 double>	24	1023
vr	<1024x1 double>	0	223.1878

Cas 1

Name	Value	Min	Max
exc	<1024x1 double>	0	5
tout	<1000x1 double>	24	1023
vr	<1024x1 double>	2.2839	223.1878

Cas 2

Name	Value	Min	Max
exc	<1024x1 double>	0	5
tout	<1000x1 double>	24	1023
vr	<1024x1 double>	0	0

Cas 3

Cas 1 : Expérience valide.

Cas 2 : Expérience non valide (**vr_min ≠ 0**). Refaire l'identification à partir de l'étape 9.

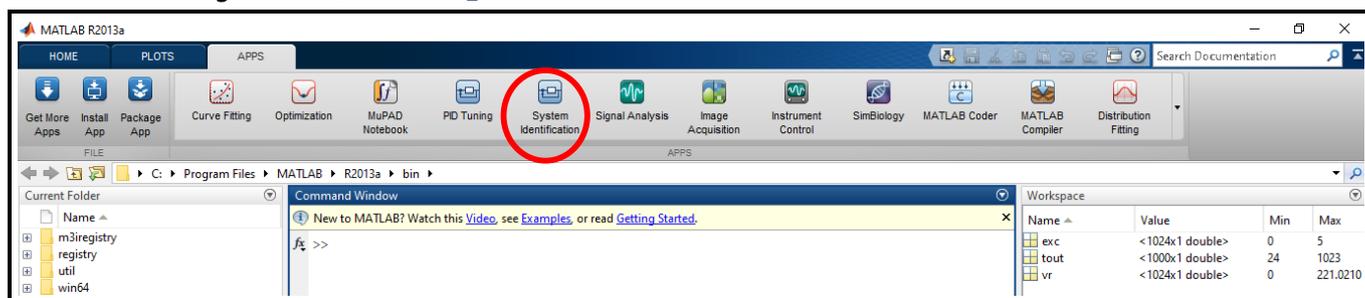
Cas 3 : Toutes les valeurs de **vr** sont nulles. Le capteur de vitesse a perdu son calibrage. Télécharger le fichier **CAPAX-MD Installation.pdf** à partir de <http://www.infoconsult.tn/fr/telechargements.html> et suivre les instructions pour son recalibrage (§ 6).

Question 1 :

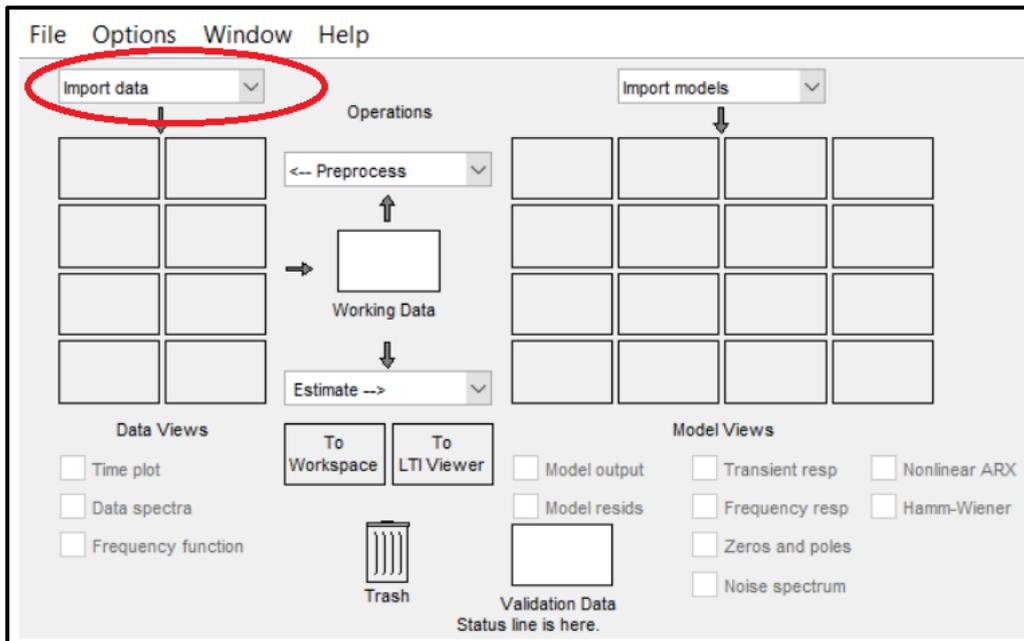
- 1- Tracer la courbe du signal d'excitation **exc** (utiliser l'instruction **Plot**).
- 2- Tracer la courbe de réponse de vitesse de rotation **vr** (utiliser l'instruction **Plot**).

3- Identification de la fonction de transfert d'un MCC et étude temporelle

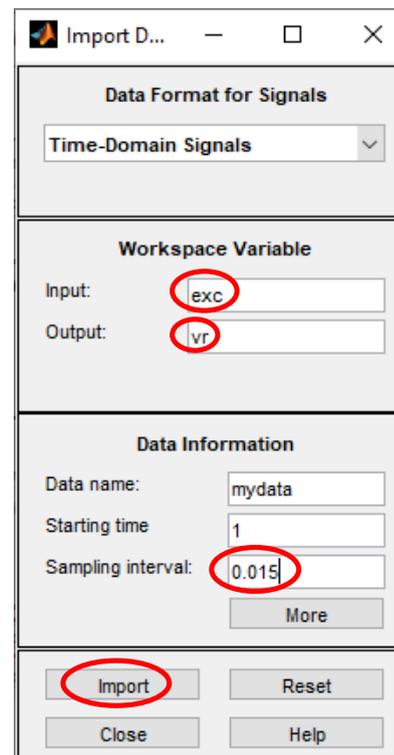
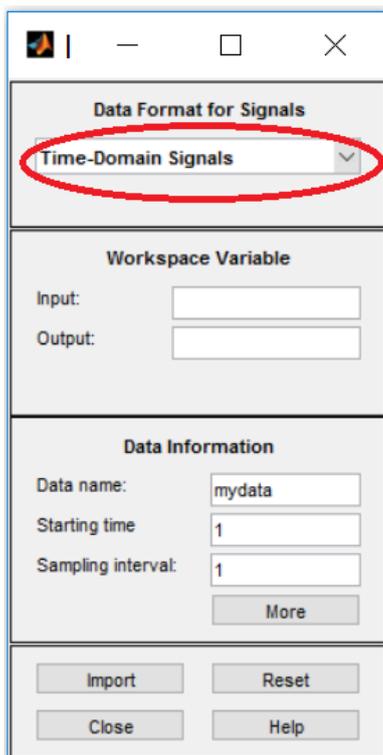
1- Dans l'onglet **APPS**, choisir **System identification**:



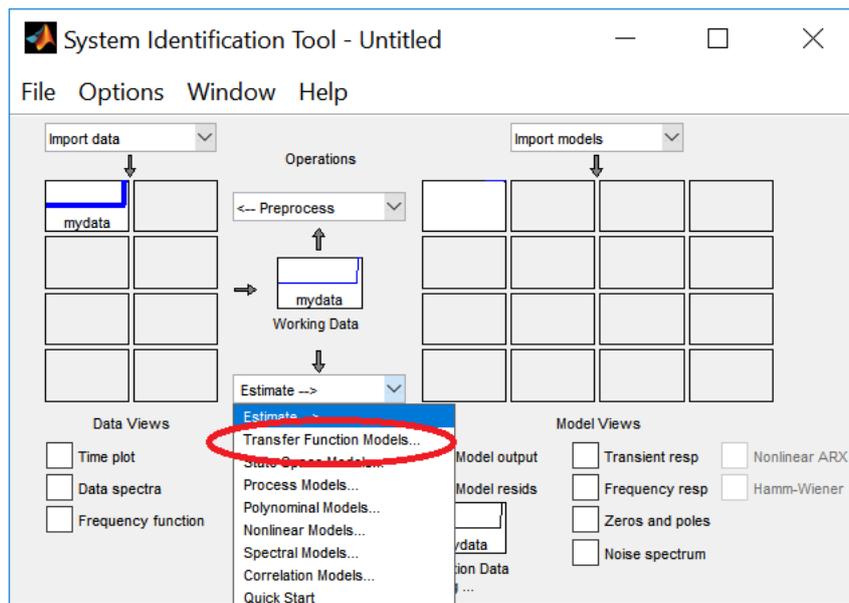
2- Dans la liste déroulante **Import data** :



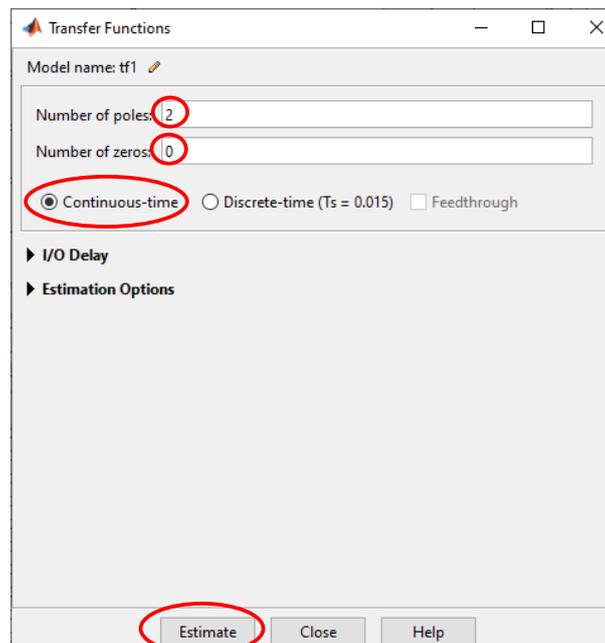
3- Choisir **Time domain data...** , remplir les champs suivants : **Input**, **Output** et **Sampling interval** et cliquer sur **Import**.



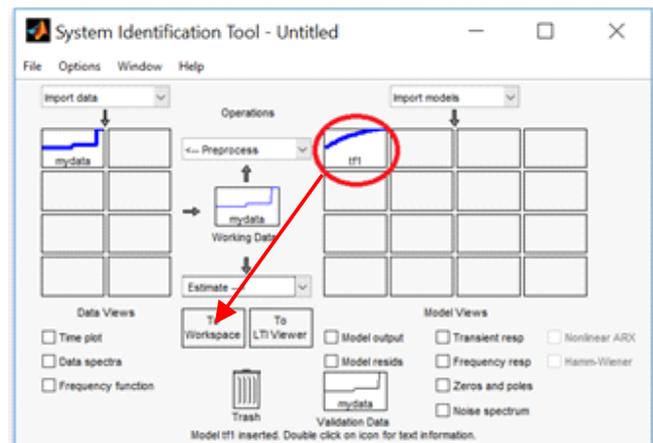
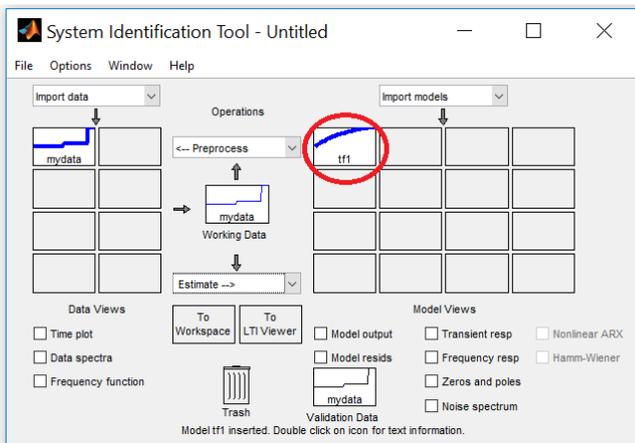
4- Dans la liste déroulante **Estimate**, choisir **Transfer Function Models**.



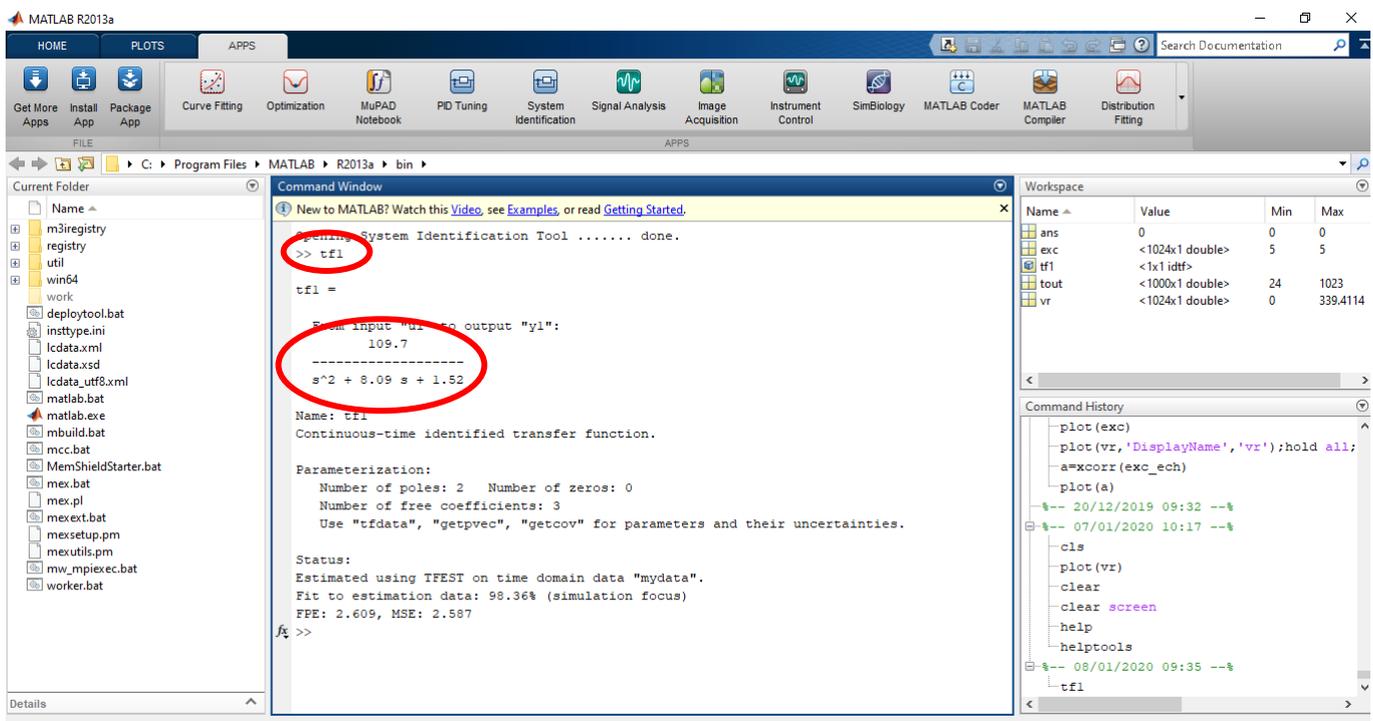
5- Remplir les champs comme indiqué dans la figure : **Number of poles**, **Number of Zeros** puis choisir l'option **Continuous-time** enfin cliquer sur **Estimate**



6- La fonction de transfert **tf1** est calculée. Cliquer sur, **tf1** maintenir le click et le faire glisser dans l'environnement **workspace** de **Matlab**.



7- Pour afficher l'équation de la fonction de transfert, taper **tf1** puis "**Entrée**" dans la fenêtre **Command Window** de **Matlab**.



Pour cet exemple

$$tf1 = \frac{109.7}{s^2 + 8.09s + 1.52}$$

Question 2:

- 1- A partir de l'environnement **workspace** de **Matlab**, déterminer la fonction de transfert continue du moteur $H(p)$ (**tf1** dans **workspace**)
- 2- Calculer le gain statique K , la pulsation propre ω_n et l'amortissement ξ du système.
- 3- Représenter la réponse indicielle du système en utilisant la commande **step** de Matlab.
- 4- Quelle est la nature du système (Amorti, oscillant, hyper amorti...)

- 5- Déterminer les pôles du système p_1 et p_2 avec deux méthodes :
 - ❖ En utilisant les expressions des pôles correspondant au régime du système.
 - ❖ En utilisant la commande **ZPK**.
- 6- Déterminer les constantes de temps τ_1 et τ_2 .
- 7- Déterminer le temps de réponse $t_r(5\%)$ du système en utilisant l'abaque du temps de réponse à 5% (Figure.3) en fonction du coefficient d'amortissement et de la pulsation propre ω_n d'un système du deuxième ordre et l'approximation suivante: $t_r(5\%) = 3 \times 2\xi\omega_n$.
- 8- Déterminer l'erreur statique en boucle ouverte $\varepsilon(\infty)$.

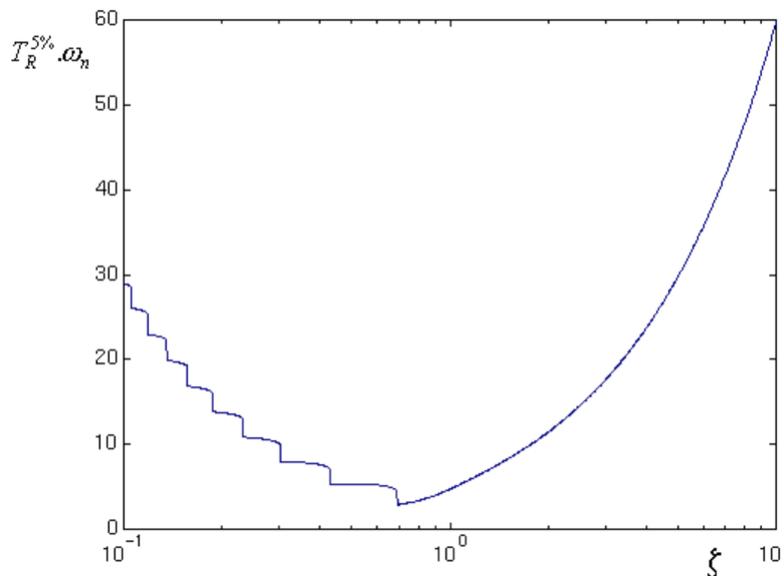


Figure 3. Abaque du temps de réponse à 5% d'un système linéaire du 2nd ordre.

4- Etude fréquentielle

On s'intéresse à étudier le comportement fréquentiel du moteur, soit $p = j\omega$.

Question 3 :

- 1- Déterminer la fonction de transfert $H(j\omega)$.
- 2- Déterminer le module $A(\omega)$ et l'argument $\phi(\omega)$ de $H(j\omega)$.
- 3- Tracer le diagramme de Bode du module et d'argument.
- 4- Déterminer les deux pulsations de coupures ω_{c1} et ω_{c2} les pentes de deux asymptotes issus de ω_{c1} et ω_{c2} .
- 5- Calculer la marge de phase et la marge du gain avec deux méthodes à l'aide du lieu de Black/Nichols de la boucle ouverte. Evaluer graphiquement la marge de gain et la marge de phase ainsi que les fréquences associées (utiliser le curseur souris pour faire apparaître les propriétés du système par l'aide contextuelle de Windows).
- 6- Comparer aux valeurs calculées par Matlab avec la commande **margin**.
- 7- Dédire sur la stabilité du système en boucle ouverte.

ANNEXES

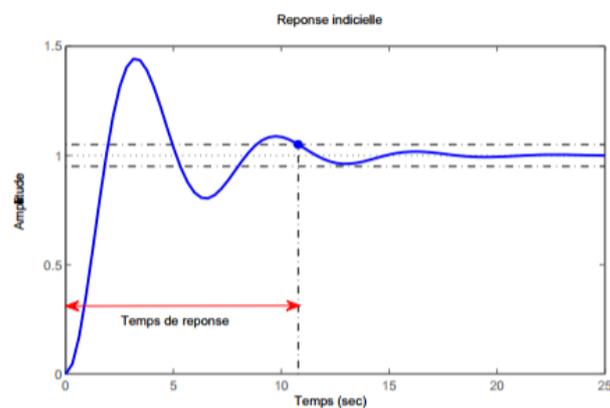
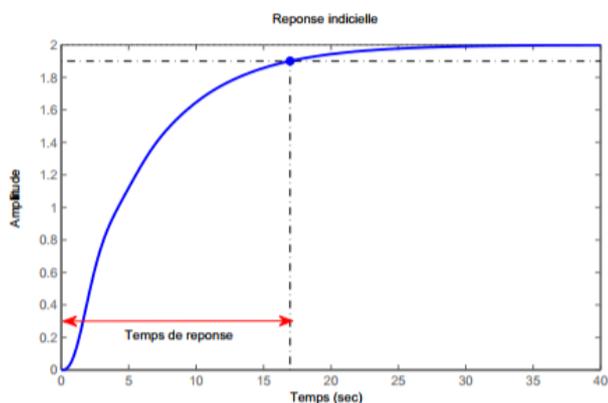
Annexe A1

Tableau 1: Paramètres et notations du moteur

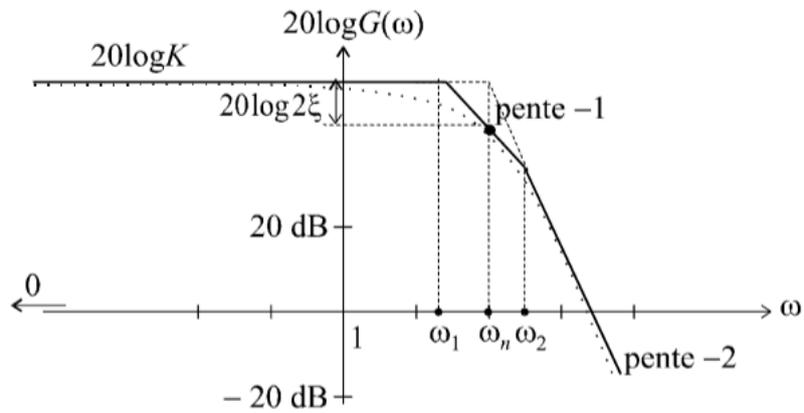
Paramètre	Définition	Unité
U	Tension aux bornes de l'induit	V
I	Courant dans l'induit	A
R	Résistance aux bornes de l'induit	Ω
L	Inductance aux bornes de l'induit	H
E	Force électro-motrice	V
J	Moment d'inertie	kgm^2
C_f	Couple de frottement	mN
f	Coefficient de frottement visqueux	mNs/rd
C_r	Couple résistant	mN
K_E	Constante de f.e.m	Vs/rd
K_c	Constante de couple	mN/A
Ω_0	Pulsation de rotation du moteur	rad/s
n_0	Vitesse de rotation du moteur	tr/mn

Temps de réponse:

Le temps de réponse d'un système est le temps mis par la sortie du système pour entrer dans la bande comprise entre $\pm 5\%$ de sa valeur finale.

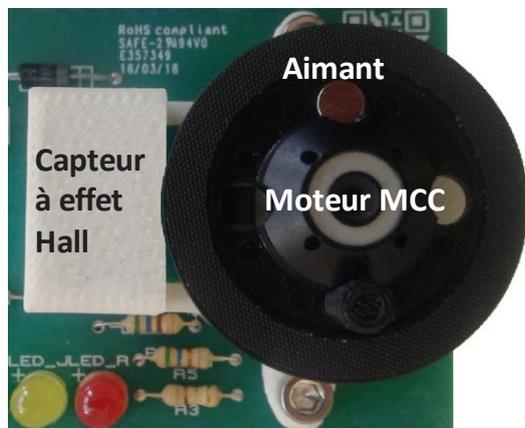
Temps de montée:

Le temps de montée d'un système est le temps mis par sa sortie pour passer de 10% de sa valeur finale à 90% de sa valeur finale.

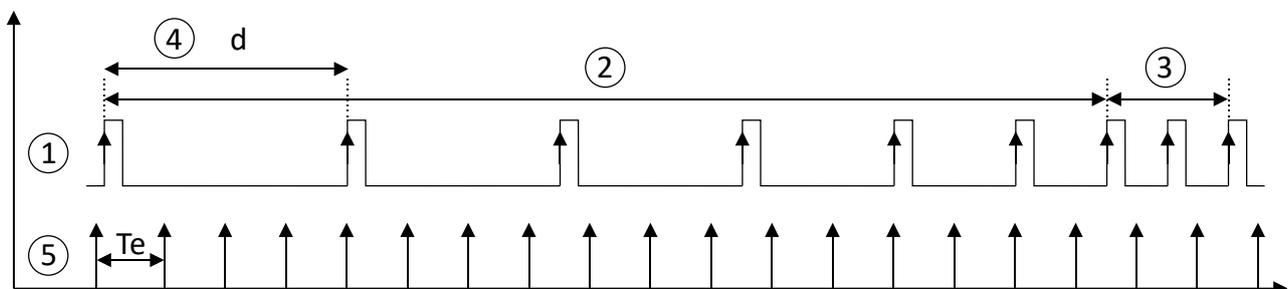
Diagramme de Bode d'un système du second ordre avec facteur d'amortissement supérieur à 1

Annexe A2

Principe de la mesure de la vitesse de rotation



A chaque passage de l'aimant au-dessus du capteur à effet Hall, celui-ci génère une impulsion. La mesure du temps séparant deux impulsions permet de calculer de la vitesse de rotation.



- ① Train d'impulsions générées par le capteur de vitesse (capteur à effet Hall).
 ② Train d'impulsions en phase d'accélération. ③ Train d'impulsions à vitesse constante.
 ④ d : Durée séparant deux impulsions. ⑤ T_e : Période d'échantillonnage.

Soit

- * d : Durée entre deux impulsions [μs]
- * Ω : Vitesse de rotation
- * T_e : Période d'échantillonnage

$$\Omega = \frac{1}{d} [\text{tr}/\mu\text{s}] = \frac{10^6}{d} [\text{tr}/\text{s}] = \frac{60 \cdot 10^6}{d} [\text{tr}/\text{mn}] = \frac{2 \cdot \pi \cdot 10^6}{d} [\text{rd}/\text{s}]$$

Exemple $d = 50\,000 \mu\text{s}$

$$\Omega = 20 [\text{tr}/\text{s}] = 1200 [\text{tr}/\text{mn}] = 125.66 [\text{rd}/\text{s}]$$

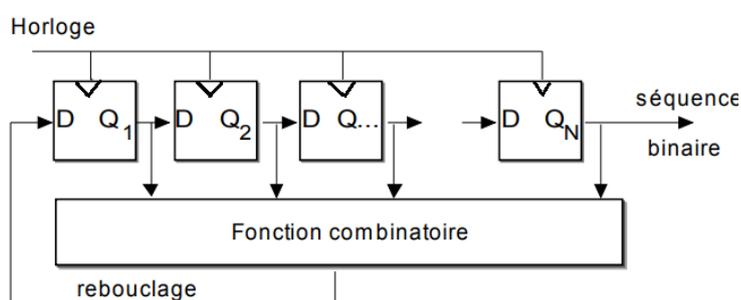
Détermination de la période d'échantillonnage maximale

D'après la notice du constructeur, $\Omega_{\text{max}} = 2830 \text{ tr}/\text{mn}$ (moteur en charge) d'où

$$d_{\text{max}} = \frac{60 \cdot 10^6}{\Omega_{\text{max}}} = 21\,201 \mu\text{s} = 21.201 \text{ ms} \rightarrow T_{e_{\text{max}}} \leq 21.201 \text{ ms}$$

Annexe A3 :

Principe de la Séquence binaire pseudo-aléatoire (SBPA)



C'est un signal binaire généré par un registre à décalage de N bits (Bascule D) rebouclé par une fonction spécifique à base de "OU exclusifs". L'état des N bascules forme un mot de N bits observé par sérialisation.

Figure 1. Registre générateur de séquence SBPA.

Si la fonction de bouclage est bien choisie, le registre peut prendre au mieux toutes les combinaisons possibles des N bits soit 2^N états différents, sauf la combinaison "0 0 0 ... 0" (le registre resterait à une valeur nulle indéfiniment). Cette structure sert à générer des mots binaires aléatoires, la séquence étant formée de $2^N - 1$ valeurs différentes du mot. A partir de la 2^N ième valeur, il y a répétition de la première valeur, le générateur "reboucle" sur sa première valeur. La séquence est donc cyclique et se répète tous les $2^N - 1$ coups d'horloge. Le signal obtenu est aléatoire et forme une *Séquence Binaire Pseudo Aléatoire*. Sa longueur maximale est donc $L = 2^N - 1$.

Tableau 1. Fonctions de rebouclage SBPA.

Nombre de cellules N	Longueur de la séquence $L = 2^N - 1$	Bits additionnés Bouclage Bi et Bj
2	3	1 et 2
3	7	2 et 3
4	15	3 et 4
5	31	3 et 5
6	63	5 et 6
7	127	4 et 7
8	255	2, 3, 5 et 8
9	511	5 et 9
10	1023	7 et 10

Pour les applications d'identification, la sortie du registre binaire présente une succession de "0" et de "1" pseudo-aléatoire, de valeur moyenne nulle, assimilable à un bruit blanc, de fonction d'autocorrélation proche d'un Dirac, $\delta(k) = 1$.

Considérons un système linéaire invariant dans le temps, de réponse temporelle $h(k)$ et de fonction de transfert $H(n)$. La relation temporelle qui relie la sortie $y(k)$ à l'entrée $x(k)$ est la suivante :

$$y(k) = x(k) * h(k)$$

La relation entre la corrélation de la sortie et l'entrée:

$$C_{yx}(k) = C_{xx}(k) * h(k)$$

Où $C_{yx}(k)$ est l'inter corrélation entre la sortie et l'entrée du filtre.

Si le signal $x(k)$ est un bruit blanc,

Alors sa fonction d'autocorrélation est une impulsion de Dirac $C_{xx}(k) = \delta(k)$, d'où :

$$C_{yx}(k) = \delta(k) * h(k) = h(k)$$

L'inter corrélation entre la sortie et l'entrée du système correspond à la réponse impulsionnelle du système.

Pour plus de détail, se référer à la référence [10].

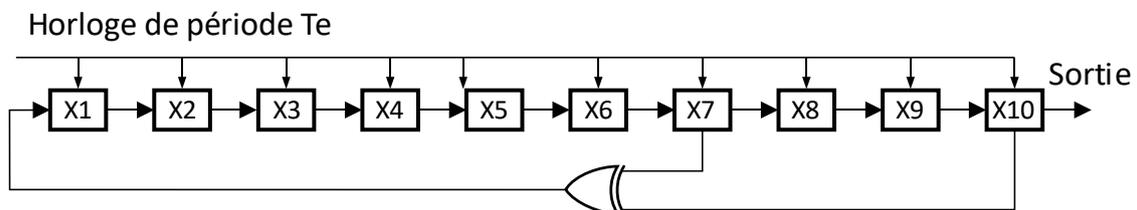


Figure 2. SBPA sur 10 bits (Principe)

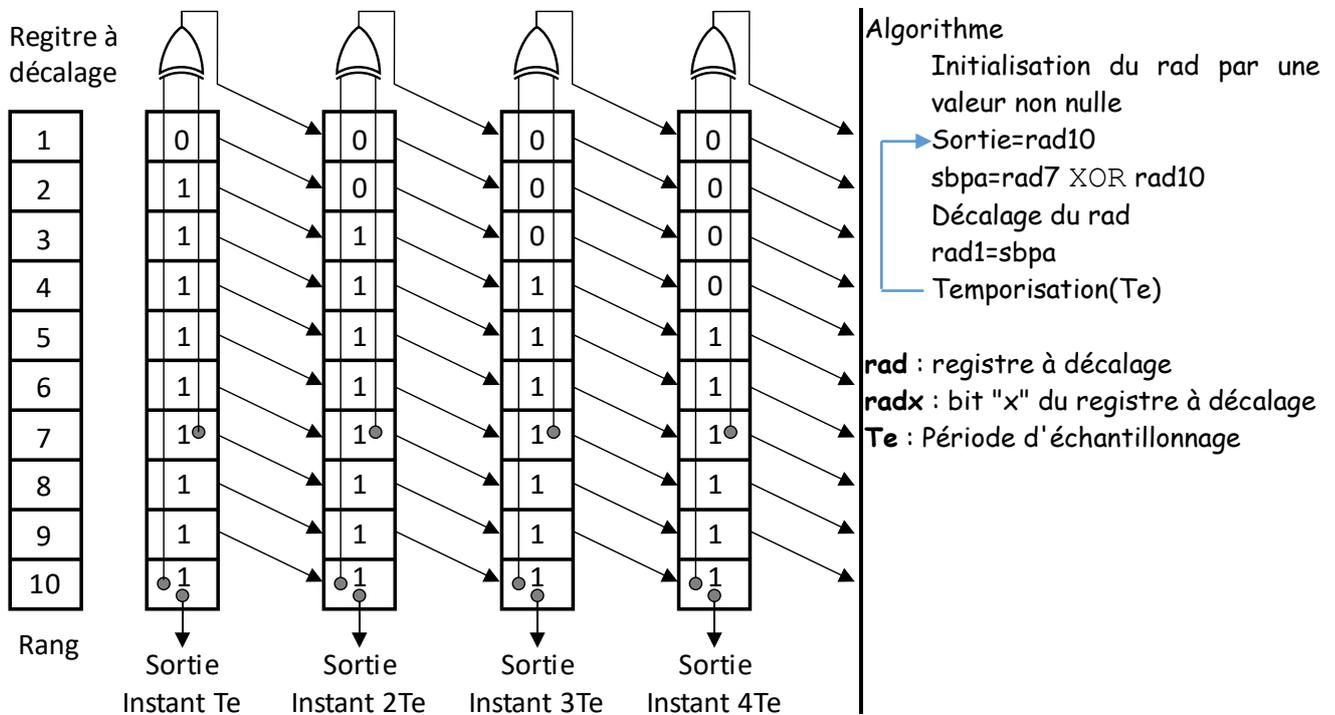
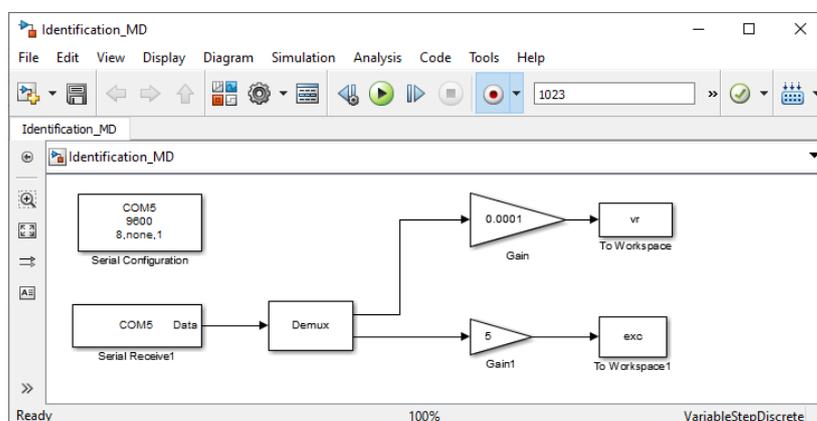
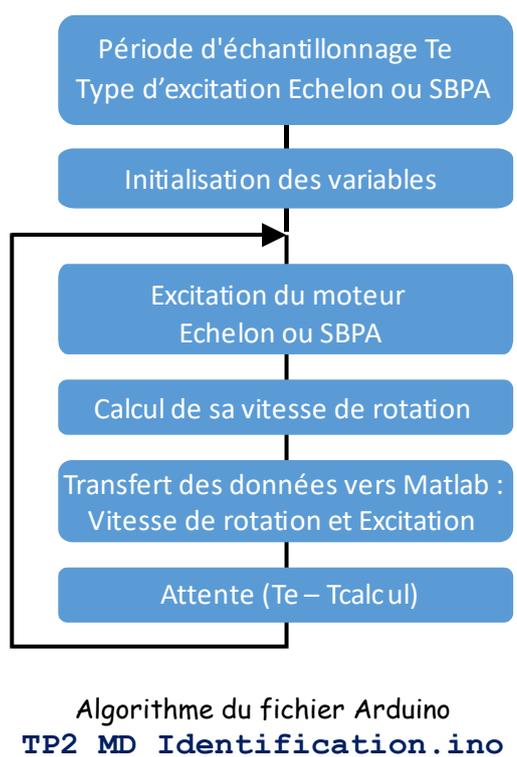


Figure 3. SBPA sur 10 bits (algorithme de calcul).

Implémentation de l'identification



Fichier SIMULINK
`Identification_MD.slx`

BIBLIOGRAPHIE

- [1] R. Ben Abdennour, P. Borne, M. Ksouri et F. M'Sahli, 2001, Identification et commande numérique des procédés industriels, Edition Technip, Paris.
- [2] D. Landau, 1988. Identification et commande des systèmes, édition Hermès.
- [3] G. B. Giannakis and E. Serpedin, 2001. A bibliography on nonlinear system identification, Signal Processing, Vol. 81, pp. 533-580.
- [4] Ljung, System Identification, Theory for the User (ISBN 0-13-881640-9)
- [5] Laszlo J. Naszodi, « On Digital Filtration in Correlation Time-of-Flight Spectrometry », Nuclear Instruments and Methods (en), vol. 161, no 1, 15 avril 1979, p. 137-140 (DOI 10.1016/0029-554X(79)90371-9).
- [6] Modélisation et identification des processus, Volume 2, Pierre Borne, Jean-Pierre Richard, ISSN 1152-0647.
- [7] Wassila Chagra, Accurate Calculation of the Settling Time of a Linear System Using New Expressions and Iterative Algorithms, Circuits Syst Signal Process, 2017.
- [8] BORNE Pierre , KSOURI Mekki Application aux procédés industriels, 1999.
- [9] <http://www.nuninger.free.fr/fichestechniques/fiche129.pdf>
- [10] https://ecitydoc.com/download/identification-par-sequences-binaires-pseudo_pdf