

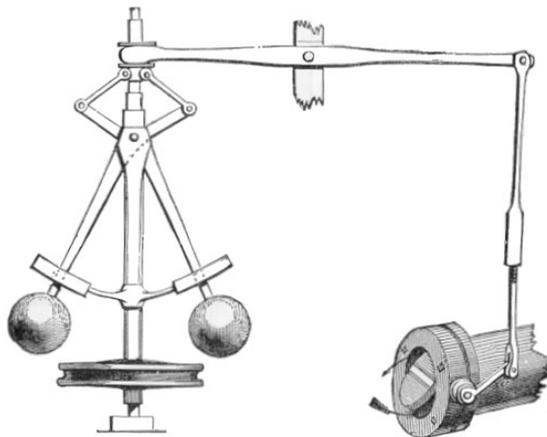


GENIE ELECTRIQUE ET INFORMATIQUE INDUSTRIELLE

Semestre 3

AUTOMATIQUE

Travaux pratiques



2022 – 2023

Sommaire et consignes

Semaines 46 et 47

TP1– (4 heures) Modélisation d’un système thermique

Semaines 48 et 49

TP2 – (4 heures) Régulation d’un système électromécanique

Les TP Automatique seront évalués à partir des comptes-rendus à remettre le 14 Décembre 2022 au plus tard à l’enseignant qui aura suivi le groupe.

TP1 – Modélisation d'un système thermique

1 – Présentation du système

Le système thermique étudié est monté sur une carte électronique de type « Arduino » comme l'illustre la figure 1. Il s'agit d'un élément chauffant, une résistance de $10\ \Omega$, $5W$, sur laquelle un capteur de température est monté (de type LM35).

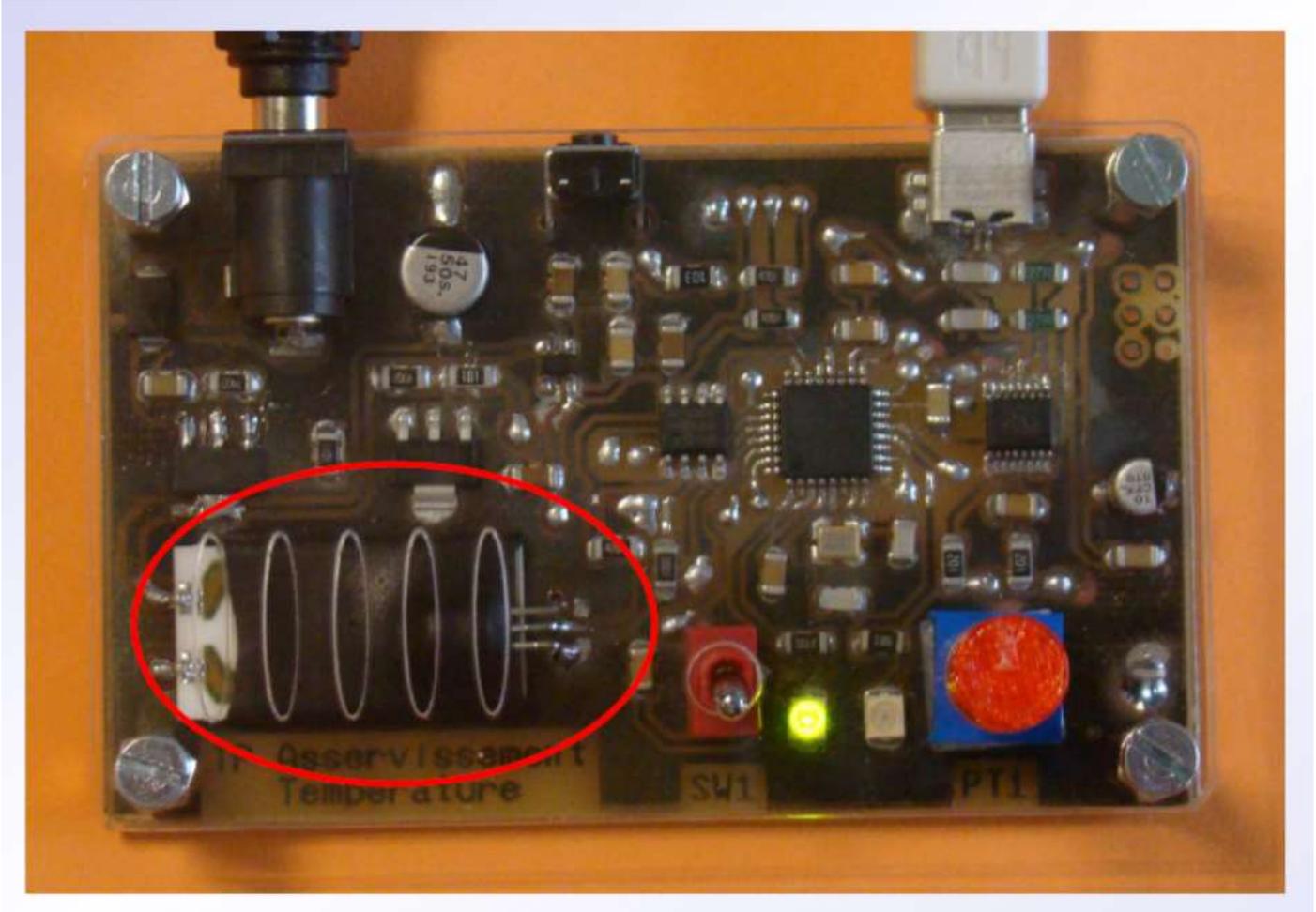


Figure 1 – Système thermique (entouré de rouge) monté sur carte électronique de type Arduino

La résistance chauffante est alimentée via un transistor MOSFET par un signal de type PWM issu de la broche 9 du microcontrôleur (commande PWM par un entier $0 < U(t) < 255$ réglant le rapport cyclique). Le capteur délivre une tension fonction de la température. Il est connecté à la broche A0 du microcontrôleur via un amplificateur opérationnel, ce qui permet une lecture de la tension image de la température via un CAN 10 bits (température convertie en $^{\circ}C$).

2 – Analyse et mise en équation

Question 1 – La chaîne de mesure de température (LM35 + AOP) apporte un gain K_{mes} égal à 0.14576 V/°C. Sachant que la pleine plage de mesure du convertisseur (CAN) est 0-5V, quelle sera la pleine plage de mesure en degrés ? Quelle sera la résolution de la mesure exprimée en degrés.

Question 2 – La résistance chauffante est alimentée via un transistor MOSFET commandé en mode PWM. On peut considérer que le transistor se comporte comme un interrupteur commandé (fermé si signal PWM à l'état haut, ouvert si signal PWM à l'état bas) et donc considérer le schéma de la figure 2.

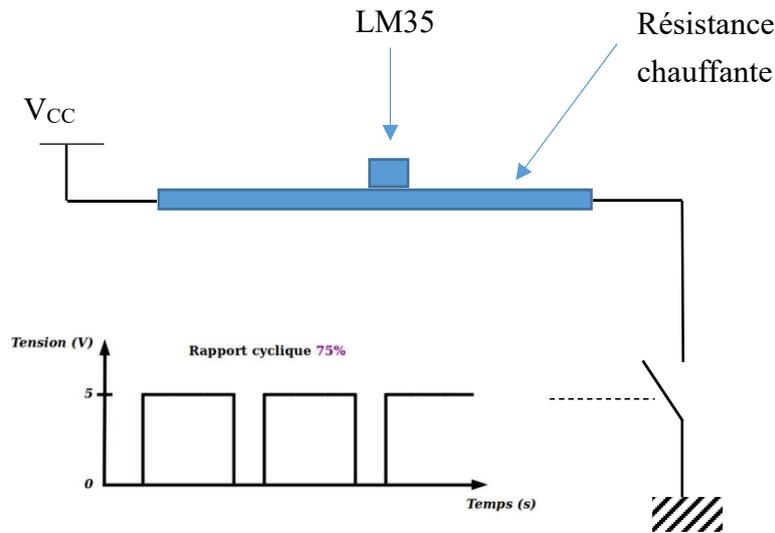


Figure 2 – Schéma d'alimentation de la résistance chauffante

Si $\alpha(t)$ désigne la valeur du rapport cyclique de la carte Arduino, donner l'expression de la puissance thermique $P_a(t)$ dissipée par la résistance en fonction de la tension d'alimentation V_{CC} et de la valeur de la résistance que l'on notera R . Pour cela, on considèrera que la résistance chauffante a une inertie thermique importante, ce qui fait qu'elle ne « voit » que la tension moyenne appliquée à ses bornes.

Question 2 – Un bilan thermique permet alors d'écrire que la puissance thermique qui participe activement à une modification de la température à l'intérieur de l'enceinte thermique correspond à la puissance apportée $P_a(t)$ moins la puissance perdue $P_{fuite}(t)$, soit :

$$P_{th}(t) = P_a(t) - P_{fuite}(t).$$

Cette puissance thermique contribue à l'élévation de la température de l'élément chauffant ce qui peut être caractérisé par l'équation différentielle :

$$C_{th} \frac{d\theta(t)}{dt} = P_{th}(t)$$

où $\theta(t)$ est la température de l'élément chauffant mesurée par le capteur et C_{th} est la capacité thermique de l'élément chauffant (mais également d'une partie de la carte électronique qui elle aussi est chauffée).

Les fuites seront supposées résulter de phénomènes de convection avec l'air ambiant, ce qui permet d'écrire :

$$P_{fuite}(t) = h_c(\theta(t) - \theta_a(t))$$

où h_c est un coefficient de convection et $\theta_a(t)$ désigne la température ambiante.

Après avoir appliqué la transformée de Laplace à l'ensemble de ces équations, construire un schéma bloc du système étudié, d'entrées $\alpha(t)$ et $\theta_a(t)$ et de sortie $\theta(t)$.

Question 3 – On notera $u(t)$ la grandeur $\alpha(t)$ ². Donner les fonctions de transfert liant :

- la température $\theta(t)$ à $u(t)$, que l'on notera $H_u(p)$,
- la température $\theta(t)$ à $\theta_a(t)$, que l'on notera $H_a(p)$.

Question 4 – Montrer que ces deux fonctions de transfert sont respectivement de la forme

$$H_u(p) = \frac{K}{1+\tau p} \quad \text{et} \quad H_a(p) = \frac{1}{1+\tau p}$$

On donnera l'expression littérale des paramètres K et τ .

3 – Identification

A ce stade, la physique a permis de déterminer la structure des modèles du système thermique étudié (fonctions de transfert $H_u(p)$ et $H_a(p)$ notamment), mais les valeurs numériques de ces paramètres ne sont pas encore connus. L'objectif de cette partie est donc d'estimer expérimentalement ces valeurs numériques. Pour cela, vous disposez de deux outils logiciels à récupérer sur Moodle :

- un programme arduino « caracterisation_v1.ino »
- un programme labview « Caracterisation.vi »

Le programme Labview supervise le fonctionnement global du système dans la mesure où il donne l'ordre au microcontrôleur d'appliquer une commande (rapport cyclique différent de 0 ou pas) en changeant l'état du switch « Etat commande » et de visualiser les signaux d'entrée (rapport cyclique) et de sortie (température) de la carte électronique (voir figure 3).

Le programme arduino permet de contrôler la carte électronique via le microcontrôleur au plus près du matériel. La valeur du rapport cyclique du signal PWM qui est appliqué au transistor MOSFET est fixée dans ce programme (à 60 pas CNA au travers de la variable COMMANDE).

Le programme Labview et le microcontrôleur communiquent via la liaison série. Pour assurer le bon fonctionnement de l'ensemble, on veillera à respecter les consignes suivantes dans l'ordre indiqué:

- connecter en premier lieu la carte électronique au PC via le cordon USB,
- téléverser le programme « caracterisation_v1.ino » et ne pas lancer le moniteur série,
- ajuster éventuellement les paramètres de communication (réglage port COM : identique à celui utilisé par la carte Arduino) du programme Labview et exécutez le.

- stopper le programme Labview uniquement avec la bouton STOP de l'application, ce qui aura pour effet de refermer correctement le port de communication,
- par la suite, en cas de soucis de communication, on pourra lancer le programme Labview « Close_com.vi ».

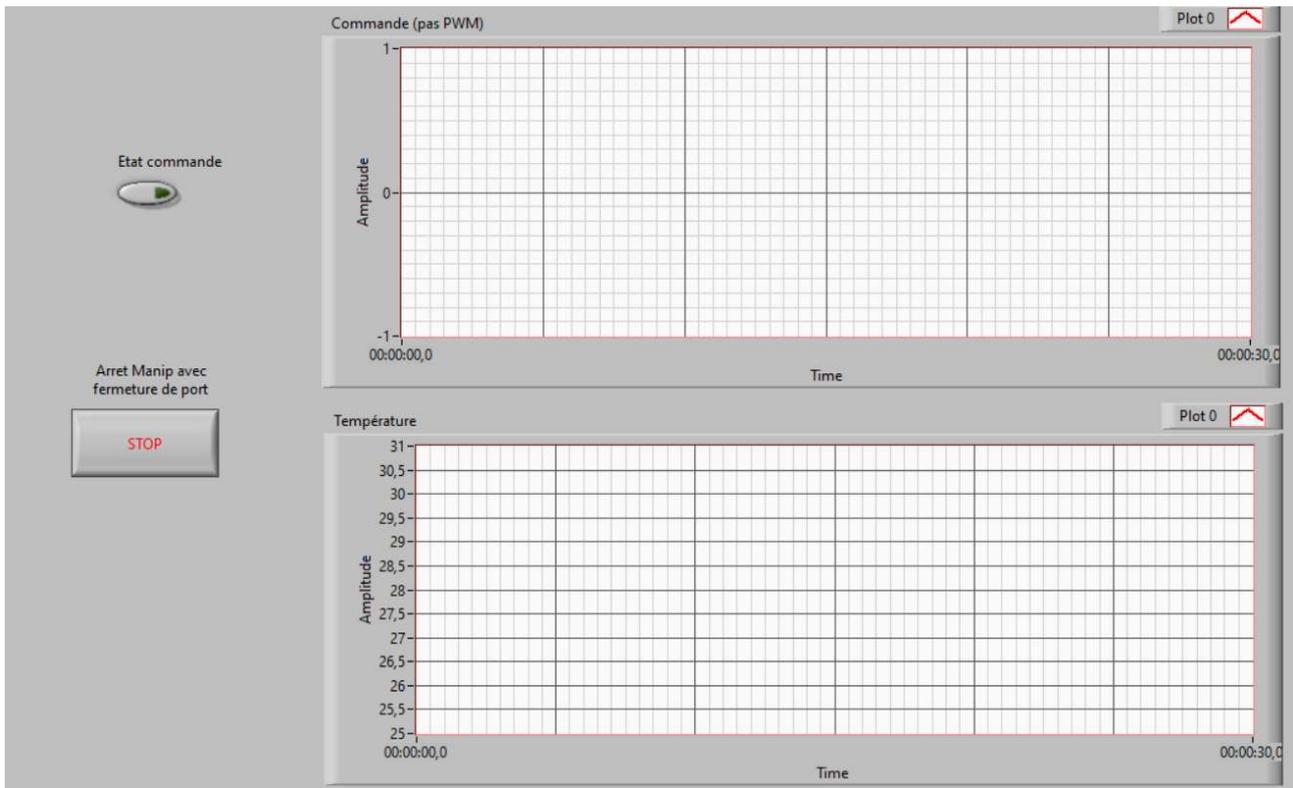


Figure 3 – Face avant du programme Labview pour la supervision.

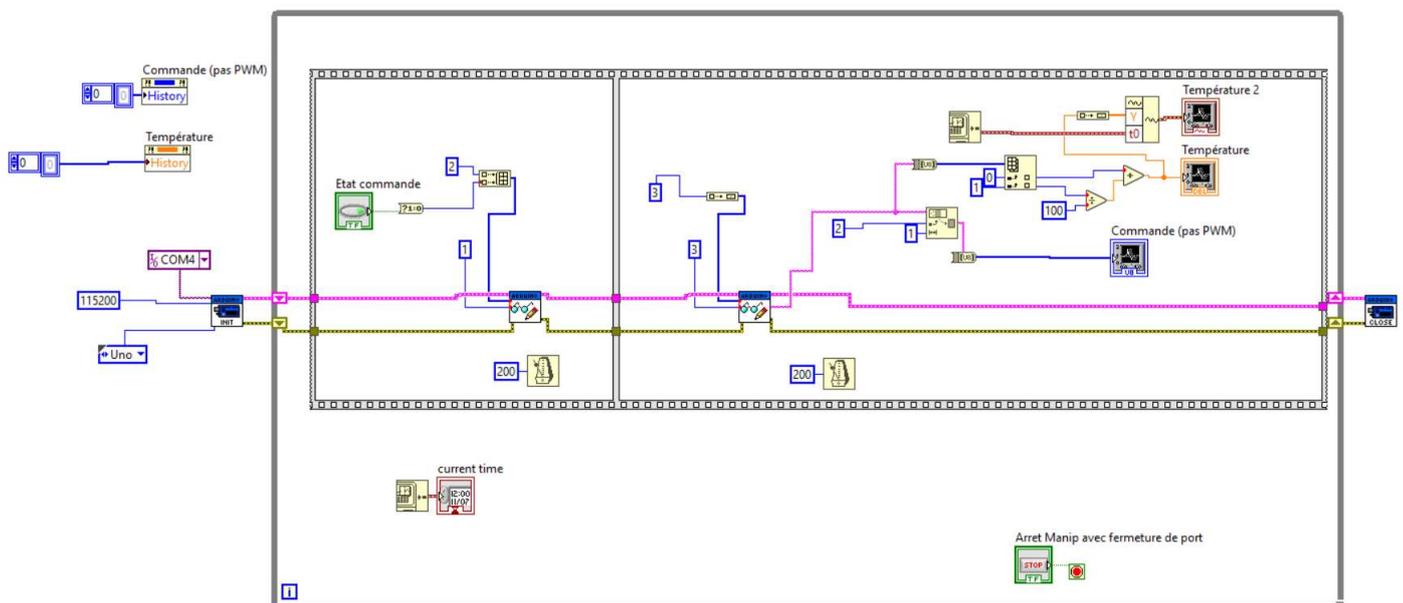
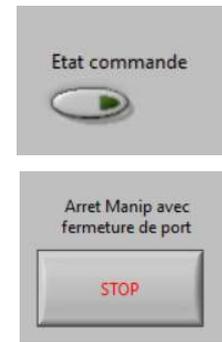


Figure 4 – Diagramme de programmation du programme LABVIEW.

Question 4 – On va à présent réaliser une première acquisition de mesures pour déterminer les valeurs numériques des paramètres K et τ . Pour cela, sans modification, effectuez les opérations suivantes :

- Téléverser le programme « caracterisation_v1.ino » et lancer le programme Labview,

- A partir de là, éviter le plus possible de faire du courant d'air sur la carte, ce qui pourrait fausser les mesures,
- Sans appliquer de commande, mesurer et noter la valeur de la température ambiante $\theta_a(t)$,
- Appliquer la commande de 60 pas CNA pour le rapport cyclique du PWM en basculant le switch « Etat commande »,
- Laisser évoluer la température pour atteindre un régime permanent (15 minutes environ).
- Stopper le programme Labview avec le bouton



Utiliser le relevé temporel obtenu pour déterminer les valeurs numériques des paramètres K et τ .

Question 5 – En utilisant le logiciel Matlab, tracer le diagramme de Bode et de Black de la fonction de transfert $H_u(p)$.

Question 6 – En utilisant la boîte à outil Simulink, implantez le schéma bloc obtenu à la question 2. Initialisez les paramètres de tous les blocs et essayez de reproduire en simulation l'expérience qui a permis d'obtenir les mesures relevées avec Labview, de façon à valider les valeurs numériques des paramètres K et τ . Relevez les graphes obtenus.

4 – Commande en boucle ouverte

Normalement, à ce stade, votre modèle du système thermique est validé. On va donc pouvoir utiliser le schéma bloc implanté sous Simulink à des fins de prédiction, notamment pour faire de la commande en boucle ouverte.

Question 7 – En utilisant la question 3, donner la transformée de Laplace du signal de température $\theta(t)$ en fonction des entrées $u(t)$ et $\theta_a(t)$.

Question 8 – Les entrées $u(t)$ et $\theta_a(t)$ étant supposées être des échelons, en utilisant le théorème de la valeur finale, donner l'amplitude du signal $u(t)$ qu'il faut imposer pour que la température en régime permanent soit de 40°C. Pour la valeur de $\theta_a(t)$, on utilisera celle trouvée à la question 4. Vérifiez avec le diagramme Simulink que la valeur imposée est la bonne et définissez la valeur du rapport cyclique $u(t)$ à imposer qui correspond.

Question 9 – Dans le programme « caracterisation_v1.ino », affecter la variable « COMMANDE » avec la valeur trouvée à la question précédente. Téléverser le programme et lancer le programme Labview. Appliquez la commande trouvée et relevez l'évolution de la température pour vérifier si effectivement la valeur de 40°C est atteinte en régime permanent. Retouchez éventuellement la valeur de « COMMANDE » pour obtenir la température demandée.

Question 10 – Utilisez un ventilateur pour souffler de l'air sur l'élément chauffant. Comment évolue la température ? Comment interprétez-vous vos observations ? On pourra s'aider du schéma Simulink construit à la question 6 en ajustant le paramètre h_c pour la température mesurée. Concluez sur l'intérêt de la boucle ouverte.

Question 11 – Prenez une autre platine disponible et refaites le même essai avec la même valeur de la variable « COMMANDE ». La valeur de la température en régime permanent est-elle de 40°C ? Comment interprétez-vous vos observations ? Concluez sur l'intérêt de la boucle ouverte.

5 – Une commande en boucle fermée pour finir

Dans cette partie, un premier essai de commande en boucle fermée va être réalisé. On va pour cela modifier le programme arduino au niveau de la partie void loop() {}. Dans cette boucle qui s'exécute en permanence avec un délai d'attente de 50ms (delay(50);), une partie du code est dédiée à la mesure de la température :

```
val = analogRead(LM35_in);  
degre = coef_LM35 * (float)val;  
degre_entiere=(int)degre;  
degre_decimale=int((degre-degre_entiere)*100);
```

A la suite de cela, on se propose de mettre en œuvre une commande dite proportionnelle.

Question 12 – A l'endroit indiqué précédemment, modifier le code de la façon suivante :

- Créer une variable consigne (consigne qui sera désignée $\theta_c(t)$ par la suite) de valeur 50 (on souhaite contrôler le système pour obtenir une température de 50°C).
- Calculer l'erreur entre cette consigne et la température mesurée : erreur = consigne – degre ;
- Multiplier par 10 (gain du correcteur proportionnel) cette erreur et affecter le résultat à la variable « COMMANDE » : COMMANDE=10*erreur.

Téléverser le programme modifié et lancer le programme Labview. Démarrer le contrôle en actionnant le switch « état commande ». Relevez les variations de rapport cyclique et de température.

Question 13 – Modifier le schéma Simulink de la question 6 pour intégrer la modification de code décrite à la question 12. Simulez le schéma et comparez les signaux obtenus à ceux relevés à la question précédente.

Question 14 – A partir du schéma bloc de la question précédente, écrire la relation liant la transformée de Laplace de la température $\theta(t)$ aux transformées de Laplace de la consigne $\theta_c(t)$ et de la température ambiante $\theta_a(t)$.

Question 15 – En utilisant le théorème de la valeur finale, déterminer la relation caractérisant le régime permanent. A partir de la mesure de température en régime permanent obtenue à la question 12, évaluer à nouveau la valeur du paramètre K de la fonction de transfert $H_u(p)$.

TP N°2 : Identification d'un moteur à partir de réponses indicielles et d'analyses harmoniques

1 Objectif

L'objectif de ce TP est de réaliser une identification temporelle et une identification fréquentielle permettant chacune d'estimer le modèle dynamique d'un Moteur à Courant Continu (MCC). On s'intéresse dans un premier temps au transfert entre le rapport cyclique du signal PWM de la Fig.1 (commande en tension de l'induit du moteur) et la sortie de l'ADC (sortie du gain k_3 , image de la vitesse de rotation du moteur). Ensuite, on déterminera la fonction de transfert $F(s)$ entre le rapport cyclique de l'induit du moteur et sa vitesse de rotation.

2 Mode d'emploi

1. Régler la tension d'alimentation aux bornes de l'induit du MCC à 20 V.
2. Lancer le logiciel de communication série sous Windows dénomé Termite.
 - Choisir le port série virtuel Com xx où xx désigne le numéro du port attribué par Windows à la carte STM32 (voir dans Gestionnaire de périphériques).
 - Configurer la communication série à 38400 bits/sec, sans parité, 1 bit de stop.
 - Cliquer sur connecter le port série sur le logiciel Termite.
3. Après le chargement du programme sur la carte STM32, l'acquisition de données temporelles se fait par deux appuis consécutifs sur BP_UP :
 - Le premier appui permet de lancer le MCC avec un rapport cyclique de 20%,
 - Le second appui permet de générer un échelon de Heaviside en entrée de 20% à 100%, de réaliser une acquisition pendant 200ms, d'arrêter le MCC, et d'envoyer les données sur le port série.
4. L'acquisition de données fréquentielles se fait par deux appuis consécutifs sur BP_DOWN :
 - Le premier appui permet de lancer le MCC avec un rapport cyclique de 60%,
 - Le second appui permet de générer un signal temporel cosinusoidal ayant un rapport cyclique variant entre 20 et 100% :

$$r(t) = \begin{cases} r_0 + r_1 \cos(\omega t) \\ 60 + 40 \cos(\omega t) \end{cases} \quad (1)$$

qui réalise une analyse harmonique entre $f = 1\text{hz}$ et $f = 100\text{hz}$, avec $\omega = 2\pi f$. Une fois l'analyse harmonique terminée, les données (fréquences, gains et déphasages) sont envoyées sur le port série et le moteur s'arrête.

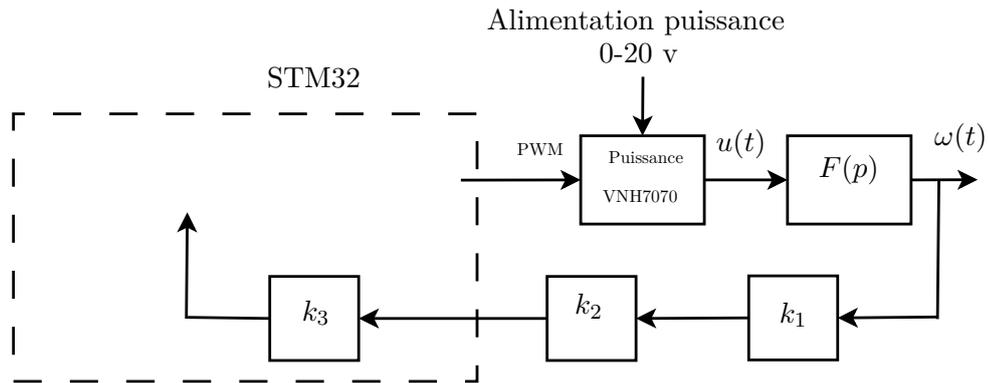


FIGURE 1 – Commande en Boucle Ouverte du MCC.

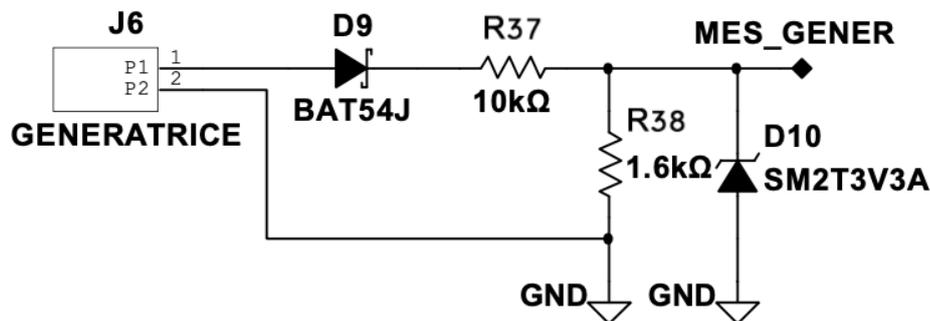


FIGURE 2 – Pont diviseur, avec GENERATRICE qui correspond à la sortie de k_1 sur la Fig.1 et MES_GENER qui correspond à la sortie du pont diviseur (sortie de k_2 sur la Fig.1).

3 Travail préliminaire

1. A partir de l'Annexe A, décrivant les caractéristiques de la génératrice tachymétrique utilisée, déterminer le gain k_1 de la force contre-électromotrice permettant de réaliser une mesure en volts de la vitesse de rotation $\omega(t)$.
2. A partir de du schéma de la Fig.2, déterminer le gain k_2 du pont diviseur.
3. Calculer le gain k_3 du convertisseur analogique numérique (ADC) 12 bits, dont la tension de référence est à 3.3v et la masse analogique est à 0v.

4 Cahier des charges

4.1 Identification et validation temporelles

Dans cette partie, on souhaite trouver le modèle mathématique du MCC à partir de données temporelles; l'entrée de l'entraînement électrique (Hacheur + MCC + Capteur de vitesse) étant le rapport cyclique du signal PWM et la sortie correspondant aux valeurs numériques de l'ADC (sortie de k_3).

4. Réaliser une série de 5 acquisitions temporelles, puis faire un copier-coller entre Termite et Matlab.
5. Créer le vecteur temps, en sachant que la période d'échantillonnage est $T_s = 1\text{ms}$.
6. Tracer sur la même figure de Matlab les 5 réponses indicielles du MCC en retranchant à chaque fois la valeur initiale. Pour cela, il est nécessaire de créer le vecteur temps avec une période d'échantillonnage de 1ms. Voir exemple de code de la Fig.3.

7. En supposant que le transfert liant la sortie de k_3 à l'entrée PWM soit modélisé par un système d'ordre 1 :

$$G(s) = \frac{K}{1 + \tau s} \quad (2)$$

estimer le gain K et la constante de temps τ du modèle.

8. Utiliser la fonction `lsim` sous Matlab pour calculer la sortie (produit de convolution) du modèle identifié avec l'échelon en entrée.
9. Superposer, aux 5 vecteurs de sortie, la sortie du modèle. Est-ce que vous validez le modèle ainsi obtenu ?
10. Déterminer $F(s)$, la fonction de transfert de l'entraînement électrique en boucle ouverte, ayant comme entrée le signal PWM et comme sortie $\omega(t)$ en tr/min.

4.2 Identification et validation fréquentielles

Dans cette partie, on souhaite trouver le modèle mathématique du MCC à partir de données fréquentielles ; l'entrée du moteur étant le rapport cyclique du signal PWM et la sortie correspondant aux valeurs numériques de l'ADC (sortie de k_3).

11. Réaliser une série de 5 acquisitions fréquentielles, puis faire un copier-coller entre Termite et Matlab.
12. Tracer sous Matlab, en haut d'une figure (`subplot(211)`) avec la fonction `semilogx`, les 5 mesures de gain superposées.
13. Tracer sous Matlab, en bas d'une figure (`subplot(212)`) avec la fonction `semilogx`, les 5 mesures de phase superposées.
14. A partir des réponses fréquentielles moyennées, déterminer les paramètres k et τ du modèle (2).
15. Superposer sur les figures précédentes (gain et phase), la réponse fréquentielle du modèle.

4.3 Validation croisée

16. Tracer sur la figure 1 :
 - (a) les 5 mesures temporelles,
 - (b) la sortie temporelle du modèle obtenu à partir de données temporelles,
 - (c) la sortie temporelle du modèle obtenu à partir de données fréquentielles,
 - (d) la sortie temporelle du modèle calculé en TD (ordre 2), et sa simplification d'ordre 1.
 - (e) quel est le meilleur modèle ?
17. Tracer sur la figure 2 :
 - (a) les 5 mesures fréquentielles,
 - (b) la sortie fréquentielle du modèle obtenu à partir de données temporelles,
 - (c) la sortie fréquentielle du modèle obtenu à partir de données fréquentielles,
 - (d) la sortie fréquentielle du modèle calculé en TD (ordre 2), et sa simplification d'ordre 1.
 - (e) quel est le meilleur modèle ?
18. Est-ce que les paramètres k et τ des modèles fréquents et temporels sont proches ?
19. Conclure.

```

1 clear, clc, close all,
2
3 % Ce code est a dupliquer sur les 5 mesures !!!
4
5 % Je colle dans V1 toutes les donnees de vitesse
6 V1 = [331, 323, 487, 674, 786, 927, 933];
7
8 % Je retranche la valeur initiale
9 V1New = V1 - V1(1);
10
11 %Je definis le vecteur temps (periode d'echantillonnage = 1e-3)
12 t = (0:(length(V1)-1)) * 1E-3;
13
14 %Je definis le signal d'entree (echelon d'amplitude 100-20 a partir de t=0)
15 u = ones(size(t)) * 80;
16
17 %Je trace la sortie V1New en fonction du temps
18 %J'identifie le Gain et la constante de temps
19
20 % Je definis la fonction de transfert identifiee
21 % Moteur = tf(); % voir help tf
22
23 % Je calcule la sortie du modele, lorsque l'entree est un echelon
24 % d'amplitude 80
25 % yModele = lsim(); % voir help lsim
26
27 % Je trace les mesures superposees a la sortie du modele
28 figure(1), plot(t, V1New) % plot(t, V1New, t, yModele)
29 xlabel('Temps (sec)')
30 ylabel('Sortie du convertisseur analogique numerique 12 bits')
31 grid on

```

FIGURE 3 – Code Matlab à compléter

Annexe A – Caractéristiques du Moteur à Courant Continu (MCC)

Tension nom.	Caractéristiques à vide		Caractéristiques au meilleur rendement				Caractéristiques au démarrage		Puiss. pic	Rés. totale	Inductance	Constante du couple	FEM
	mA	Rpm	g-cm	Rpm	mA	%	g-cm	mA	Watts	Ohms	mH	Kg-Cm/A	V/1000 trs
24	105	3750	392	3272	736	74.6	3073	5050	29.6	4.75	3.93	0.6214	6.28

FIGURE A.1 – Fiche signalétique du MCC

Annexe B – Modalité de calcul du gain et du déphasage lors de l'analyse harmonique

Lorsque le dispositif d'entraînement électrique est soumis au signal cosinusoidal (1), la sortie se comporte comme :

$$y(t) = y_0 + A_\omega \cos(\omega t - \varphi_\omega) \quad (\text{B.1})$$

$$= y_0 + A_\omega \cos(\omega t) \cos(\varphi_\omega) + A_\omega \sin(\omega t) \sin(\varphi_\omega) \quad (\text{B.2})$$

En calculant le produit scalaire de $y(t)$ par $\cos(\omega t)$ et $\sin(\omega t)$, on obtient respectivement :

$$\langle y(t), \cos(\omega t) \rangle = y_0 \langle \mathbf{1}, \cos(\omega t) \rangle + A_\omega \cos(\varphi_\omega) \langle \cos(\omega t), \cos(\omega t) \rangle + A_\omega \sin(\varphi_\omega) \langle \sin(\omega t), \cos(\omega t) \rangle \quad (\text{B.3})$$

$$\langle y(t), \sin(\omega t) \rangle = y_0 \langle \mathbf{1}, \sin(\omega t) \rangle + A_\omega \cos(\varphi_\omega) \langle \cos(\omega t), \sin(\omega t) \rangle + A_\omega \sin(\varphi_\omega) \langle \sin(\omega t), \sin(\omega t) \rangle \quad (\text{B.4})$$

où $\mathbf{1} : t \mapsto 1$ désigne la fonction identiquement égale à 1 et où le produit scalaire est calculé sur un nombre entier de périodes $N \in \mathbb{N}^*$:

$$\langle f(t), g(t) \rangle = \oint_{NT} f(t)g(t)dt \quad (\text{B.5})$$

où $T = \frac{2\pi}{\omega}$ désigne une période.

Il s'ensuit, compte-tenu de l'orthogonalité des fonctions sinus et cosinus,

$$\langle y(t), \cos(\omega t) \rangle = A_\omega \cos(\varphi_\omega) \frac{NT}{2} \quad (\text{B.6})$$

$$\langle y(t), \sin(\omega t) \rangle = A_\omega \sin(\varphi_\omega) \frac{NT}{2} \quad (\text{B.7})$$

soit

$$A_\omega^2 = \frac{4}{N^2 T^2} \left(\langle y(t), \cos(\omega t) \rangle^2 + \langle y(t), \sin(\omega t) \rangle^2 \right) \quad (\text{B.8})$$

$$\varphi_\omega = \begin{cases} \arctan \left(\frac{\langle y(t), \sin(\omega t) \rangle}{\langle y(t), \cos(\omega t) \rangle} \right) & \text{si } \langle y(t), \cos(\omega t) \rangle > 0 \\ \pi + \arctan \left(\frac{\langle y(t), \sin(\omega t) \rangle}{\langle y(t), \cos(\omega t) \rangle} \right) & \text{si } \langle y(t), \cos(\omega t) \rangle < 0 \end{cases} \quad (\text{B.9})$$

En divisant l'amplitude de la sortie par l'amplitude de l'entrée, on obtient le gain de la fonction de transfert à chaque pulsation :

$$G_\omega = A_\omega / A_0 \quad (\text{B.10})$$

Le déphasage de la fonction de transfert à chaque pulsation est quant à lui donné par (B.9).