Rapport TP1 CSCL

BÉNISTANT Raphaël, BOURLOT Xavier, 3IMACS-AE-C, Binôme 2

30 mai 2020

Table des matières

1 Introduction

Le but de cette manipulation est de modéliser un système électromécanique à partir d'une analyse fréquentielle, ainsi qu'à partir d'une analyse temporelle. Nous utiliserons pour cela le logiciel MATLAB.

2 Identification des paramètres K_s et K_g

2.1 Manipulation

Dans cette 1$^{\text{ère}}$ partie l'objectif est de déterminer les coefficients K_s et K_g. Afin de pourvenir à notre résultat, nous allons tracer respectivement la caractéristique (θ_s,V_s) et la caractéristique ($\dot{\theta}$,V_g). Le relevé des valeurs correspond respectivement aux ligne 5, 6 et 19, 20 dans le code MATLAB.

2.2 Calcul

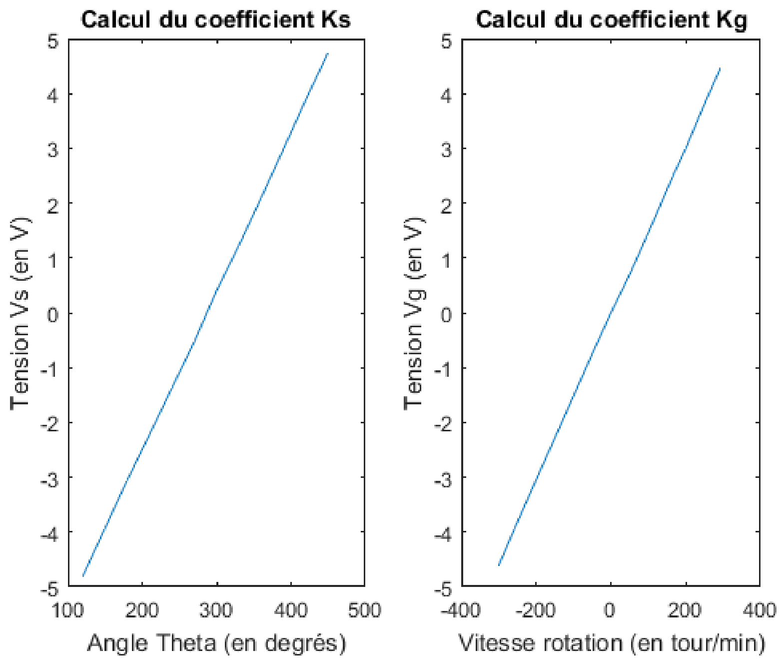

FIGURE 1 – K_s et K_g

Les pentes de chacune droite correspondent respectivement aux valeurs des coefficients K_s et K_g à des constantes près qui sont spécifiées lignes 17 et 30 du code MATLAB.

Les valeurs des coefficients sont notées dans le tableau suivant :

$K_s(V/rad)$	$K_g(Vs/rad)$
1.6571035	0.0143404

3 Identification du moteur dans le domaine temporel

3.1 Manipulation

Afin de caractériser entièrement le moteur dans le domaine temporel, nous avons mesuré la réponse indicielle de $\dot{\theta}$ à l'oscilloscope. Nous avons donc appliqué au système un échelon d'amplitude pic-à-pic 6V.

3.2 Calcul

La valeur de $\dot{\theta}$ quand le sytème est stabilisé, c'est-à-dire quand $\dot{\theta}$ vaut K_m, est multiplié par le coefficient K_g et par l'amplitude pic-à-pic de l'échelon qui n'est pas unitaire. Il faut corriger le résultat grâce à une constante précisée à la ligne 33 du code

MATLAB.
La valeur de la constante de temps se lit sur l'axe des temps lorsque $\dot{\theta}$ atteint 63% de sa valeur finale.
Les valeurs des constantes sont regroupées dans le tableau suivant :

$K_m(s.u.)(approx.)$	$T_m(ms)(approx.)$
61.016391	308

4 Identification du moteur dans le domaine fréquentiel

4.1 Manipulation

Pour identifier le moteur dans le domaine fréquentiel, nous avons effectué un relevé des amplitudes et des phases de sortie de $\dot{\theta}$ pour des entrées sinusoïdales d'amplitude pic-à-pic 4V et de fréquences variables. Ce relevé est observable de la ligne 35 à 39 du code MATLAB. Avec ces données nous avons pu tracer les diagrammes de Bode et de Nyquist correspondant. La constante de temps τ, calculée lors l'analyse dans le domaine fréquentielle, est de 308ms soit une fréquence de coupure ω_n d'environ 1,935rad/s. Nous avons de ce fait pris plus de points autour de cette fréquence de coupure.

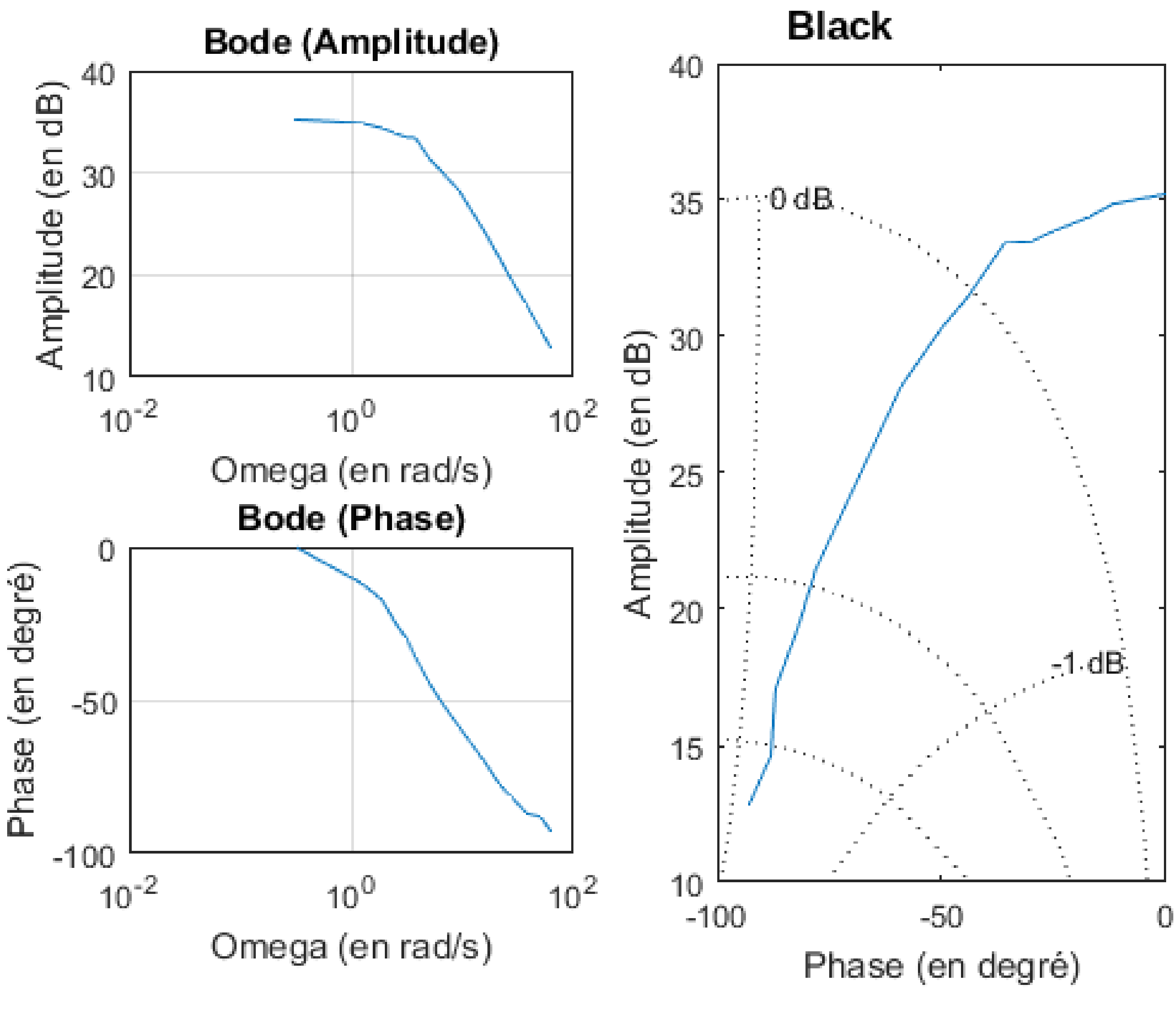

FIGURE 2 – Bode et Nyquist

Il n'est pas possible de réaliser l'analyse fréquentielle directement sur θ car l'expression de sa fonction de transfert contient un intégrateur et comme $\lim_{t\to\infty} \dot{\theta}(t) \neq 0$ alors $\lim_{t\to\infty} \theta_s(t)$ diverge.

4.2 Calcul

Le gain statique K_m se lit aux niveaux des basses fréquences. Il ne faut pas oublier de convertir le gain en dB en gain sans unité. Il est également possible de déterminer la constante de temps T_m grâce à la pulsation de coupure ω_n, c'est-à-dire la pulsation pour laquelle on enregistre une perte de 3dB par rapport au gain statique, grâce à la formule suivante :

$$T_m = \frac{1}{\omega_n}$$

Les valeurs calculées sont notéés dans le tableau suivant :

$K_m(s.u.)$	$T_m(ms)$
57.676646	198.9258

5 Modélisation sous Simulink

La modélisation Simulink de notre système permet de valider la valeurs des constantes déterminées tout au long de nos manipulations. Notre modèle Simulink ne prend en compte que la fonction de transfert :

$$\frac{\dot{\theta}}{V_e} = \frac{K_m}{1+T_m s}$$

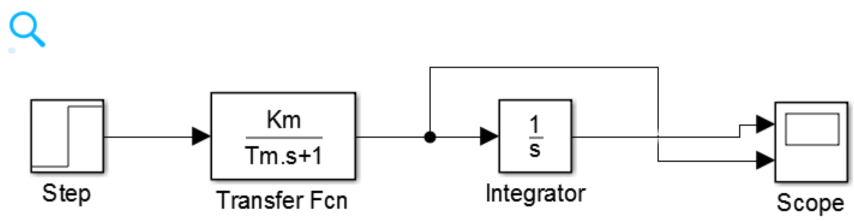

FIGURE 3 – Simulink block diagram

Il faut donc multiplier la sortie $\dot{\theta}$ par K_g pour retrouver les résultats observables lors des manipulations.

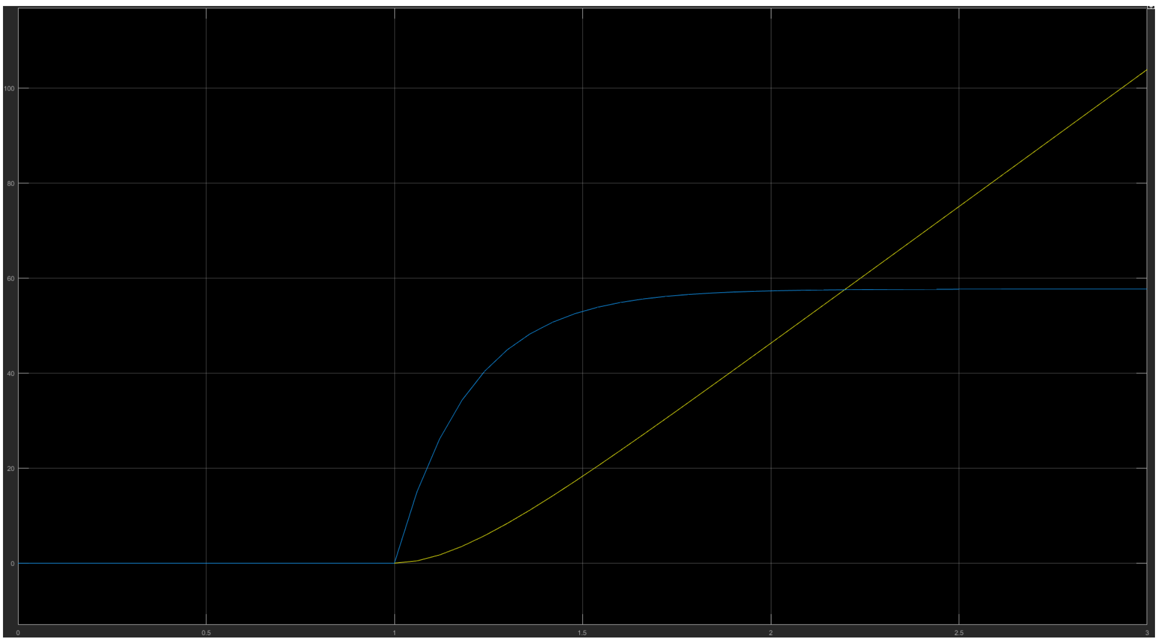

FIGURE 4 – Impulse response of the system

6 Conclusion

Ainsi, les différentes mesures des capteurs du système ont permis de caractériser les gains relatifs à la mesure de l'état, et l'observation fréquentielle du système a permis d'élaborer un modèle proche de la réalité. La modélisation sous Simulink offre une réponse temporelle semblable au comportement du système réel, on pourra donc calculer une loi de commande asservissant correctement le système.

A Annexes

A.1 Code Matlab

breaklines

```
clc
clear all
close all

theta=linspace(120,450,12);
Vs=[-4.81 -3.93 -3.05 -2.22 -1.37 -.53 .4 1.23 2.10 2.99 3.89 4.75];

figure;
subplot(121);
plot(theta,Vs);
xlabel('Angle Theta (en degres)');
ylabel('Tension Vs (en V)');
title('Calcul du coefficient Ks');

res=polyfit(theta,Vs,1);

Ks=res(1)*180/pi

omega=[-300 -258 -197 -149 -99 -47 0 52 96 166 200 252 293 ];
Vg=[-4.61 -3.94 -3 -2.27 -1.5 -0.7 0 0.73 1.4 2.5 3 3.84 4.47 ];

subplot(122);
plot(omega,Vg);
xlabel('Vitesse rotation (en tour/min)');
ylabel('Tension Vg (en V)');
title('Calcul du coefficient Kg');

res=polyfit(omega,Vg,1);

Kg=res(1)*9*2*pi/60

tau=.308;
Km=5.25/(Kg*6)

freq=[0.05 0.2 0.3 0.4 0.5 0.6 0.8 1 1.5 2.5 3.5 4.5 6 8 10]*2*pi;
Phase=-[0 12 17.28 25 30 36 44.4 50 59 70 78 82 87 88 93];
Amplitude=[3.31 3.17 3 2.83 2.71 2.69 2.13  1.88 1.47 0.94 0.68 0.53 0.41 0.309 0.25]/(4*Kg);

Amplitude=20*log10(Amplitude);

figure;
subplot(221);
semilogx(freq,Amplitude);
grid on;
xlabel('Omega (en rad/s)');
ylabel('Amplitude (en dB)');
title('Bode (Amplitude)');

subplot(223);
semilogx(freq,Phase);
grid on;
xlabel('Omega (en rad/s)');
ylabel('Phase (en degre)');
```

```
title('Bode (Phase)');

Km=10^(35.22/20)
Tm=1/5.027

subplot(2,2,[2 4]);
plot(Phase,Amplitude);
ngrid;
xlabel('Phase (en degre)');
ylabel('Amplitude (en dB)');
title('Nyquist');
```