

Outils numériques : Modélisation et commande du Robot Scara

Boris Alexandre de Baudel - ENS Rennes

January 31, 2024

Abstract

Le sujet de ce travail pratique consiste à modéliser et à analyser la commande d'un Robot Scara. Pour cela, nous avons utilisé plusieurs outils de modélisation notamment à l'aide du logiciel Matlab sur la partie Simulink. De même, sur la partie mécanique du robot, les modules du robot SolidWorks ont aussi été nécessaires pour l'étude. Ainsi, l'étude se répartie en 5 parties : La présentation de Matlab et du Robot, la commande du moteur à courant continu, la modélisation mécanique du robot, la commande en vitesse du robot et, finalement, la commande en position. Voici à quoi ressemble le robot :



Figure 1: Image du Robot Scara.

1 Présentation de l'outil Matlab Simulink

Question 1 :

Pour cette question, nous devons modifier notre courbe de sortie afin qu'elle soit le plus en accord avec les données expérimentales. Avec la commande "preprocess" on trace ainsi la courbe en sortie.

Par identification, il s'agit d'un système du deuxième ordre. Par ailleurs, le système nous donne les valeurs de K et de T des données. On a :

$$T = 900K = 9,99$$

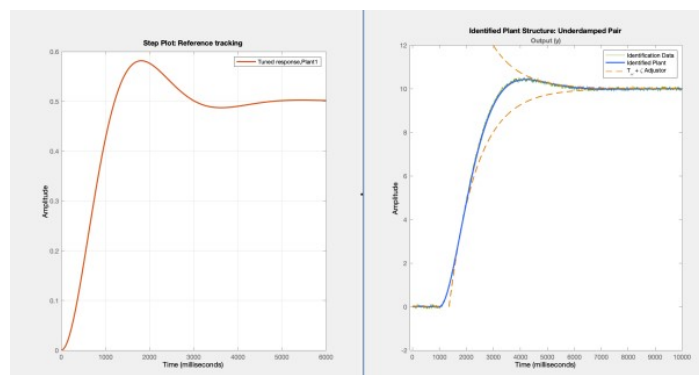


Figure 2: Valeur de sortie dans la simulation du modèle.

Question 2:

On rajoute ainsi un correcteur PID et on observe certaines variations dans notre diagramme de Bode. En effet, le caractère intégrateur, s'il s'agit d'un PI va augmenter la précision et diminuer la stabilité et la rapidité. Puis s'il on choisit un proportionnel dérivé on augmente ainsi la rapidité et la stabilité du système. Finalement, augmenter le coefficient K augmente la stabilité et la rapidité. Voici deux cas de figures pour des valeurs différentes du correcteur :

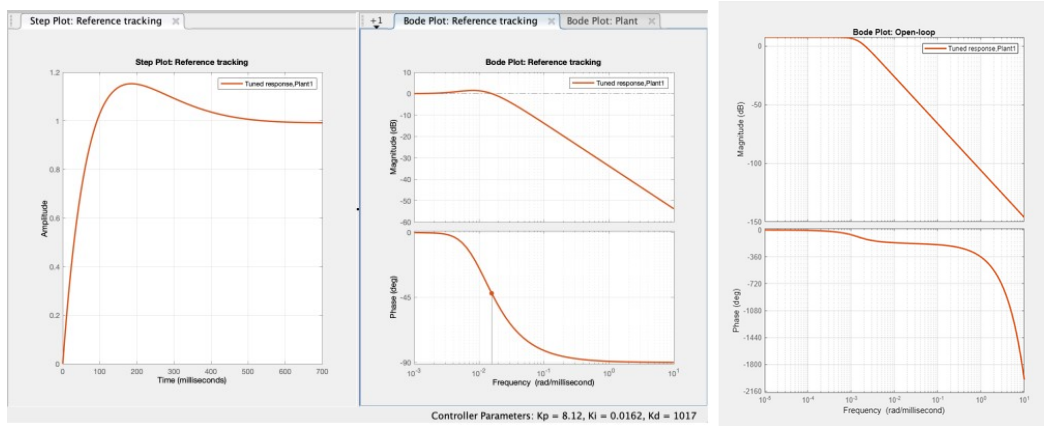


Figure 3: Diagramme de Bode pour deux valeurs différentes.

2 Modèle du moteur à courant continu

Question 3 :

Voici le modèle mécanique et électrique présenté par Matlab Simulink que l'on va utiliser pour modéliser le moteur :

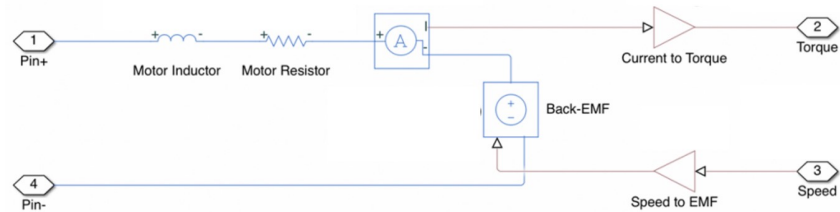


Figure 4: Modèle électrique du moteur.

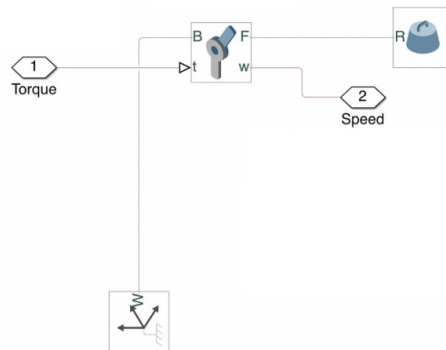


Figure 5: Modèle mécanique.

Le modèle du moteur à courant continu est modélisable par un circuit RL en serie muni d'un générateur. D'après la loi des mailles on a :

$$U(t) = E(t) + R * I(t) + L \frac{di}{dt}$$

De même, d'après le théorème du moment dynamique appliquée à l'arbre moteur, on a :

$$J \frac{d(\Omega(t))}{dt} = Cm(t) + Cr(t)$$

Finalement, d'après les relations de l'induction magnétiques et la relation entre le couple moteur et l'intensité, on a :

$$E(t) = Kv * \Omega(t)$$

$$Cm(t) = Kc * i(t)$$

Question 4 :

Pour déterminer les constantes de temps électriques nous allons observer la sortie en vitesse et en courant du moteur face à une entrée de type échelon. Il est intéressant de remarquer que ,pour la courbe du courant, le rotor du moteur à courant continue est bloqué.En effet, en prenant la valeurs à 0,63 de la valeur finale, on observera le temps caractéristique en abscisse (temps pour que le regime permanent soit atteint). On observe à la sortie on obtient :

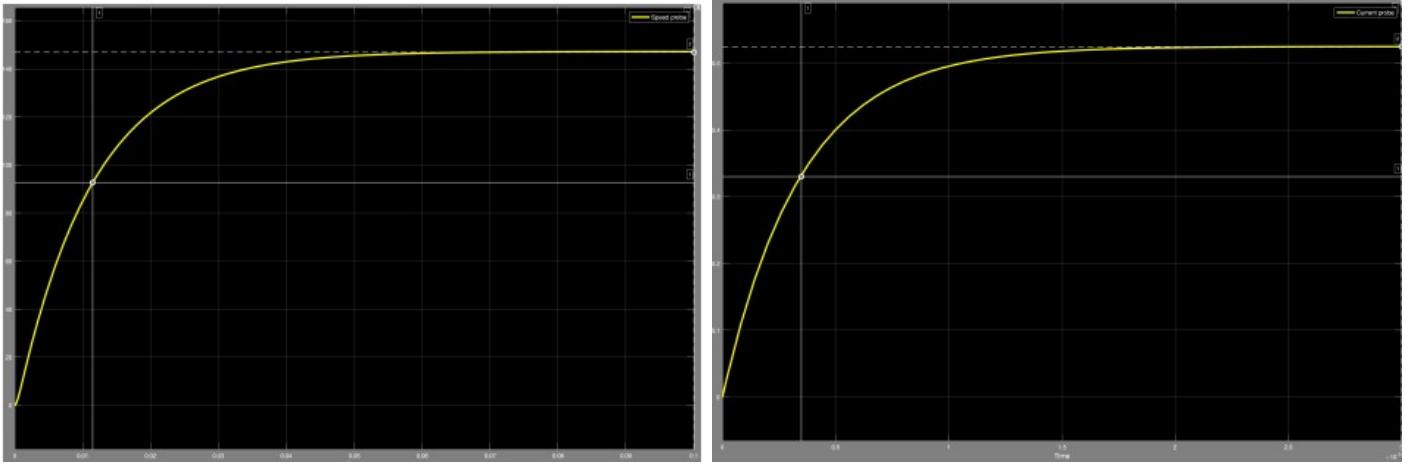


Figure 6: Vitesse de sortie et courant du moteur en fonction du temps.

On peut relever de cette manière les valeurs :

$$\tau_{meca} = 10ms$$

$$\tau_{elec} = 0,34ms$$

Ainsi, pour trouver la valeurs de f , en régime permanent, on utilise les equations du moteur à courant continue et on a :

$$J \frac{d(\Omega(t))}{dt} = Cm(t) + Cr(t) = Cm(t) + f\Omega(t)$$

$$0 = Kc * I_{permanent} + f * \Omega_{permanent}$$

$$f = \frac{Kc * I_{permanent}}{\Omega_{permanent}} = 0,0049 N.mrad^{-1}$$

$$J = f * \tau_{meca} = 0,000049 Kg.m^2$$

Question 5 :

Le programme établie est une optimisation de la valeur L de l'inductance et de R la résistance. Celle-ci va choisir des valeurs pour quelles soient le plus proche possible des valeurs expérimentales.

L_opt = 3.200314e-03	$S(\theta) = \sum_{i=1}^N (y_i - f(x_i; \theta))^2 = \sum_{i=1}^N r_i^2(\theta)$
R_opt = 9.830119e+00	
Kc_opt = 1.987115e-02	
Kv_opt = 4.004616e-02	

Figure 7: Résultats de l'optimisation et formules des moindres carrés .

Pour cela, il y a une fonction "cout" qui évalue, à l'aide de la méthode des moindres carrés, la somme de tous les écarts aux carrés. Ainsi, le programme va calculer les écarts et va ainsi choisir des valeurs de L et R pour minimiser S, la somme des écarts.

Question 6 :

Pour optimiser deux paramètres supplémentaire Kc et Kv on va recopier les lignes précédentes de R et L mais on va remplacer ceci par Kc et Kv dans l'ensemble du programme pour optimiser Kc et Kv par la même methode des moindres carrés pour Kc et Kv.

```
function optimisation(t,i)

% paramètres initiaux
L_init = 3.3e-3; % Henry
R_init = 12; % Ohm
Kv_init = 0.03; % V.rad^-1.s
Kc_init = 0.03; % N.m.A^-1

param_init = [L_init,R_init,Kv_init,Kc_init];

global g_t g_i;
g_t = t;
g_i = i;

set_param('motoreducteur','SimMechanicsOpenEditorOnUpdate','off');

cout_init = calcul_cout(param_init);
fprintf('cout initial = %d\n',cout_init);

%param_opt = param_init;
options = optimset('Display','iter', ...
    'MaxIter',1000, ...
    'MaxFunEvals',1000, ...
    'PlotFcns',@optimplotx, ...
    'TolFun',1e-3, ...
    'TolX',1e-3);

function cout = calcul_cout(param)
global g_t g_i simout;

fprintf('L = %d\n',param(1));
fprintf('R = %d\n',param(2));
fprintf('Kc = %d\n',param(3));
fprintf('Kv = %d\n',param(4));

set_param('motoreducteur/DC motor','L',sprintf('%g',param(1)));
set_param('motoreducteur/DC motor','R',sprintf('%g',param(2)));
set_param('motoreducteur/DC motor','Kc',sprintf('%g',param(3)));
set_param('motoreducteur/DC motor','Kv',sprintf('%g',param(4)));

sim('motoreducteur',g_t(end));

t_sim = simout.time;
i_sim = simout.signals.values;

i = interp(t_sim,i_sim,g_t);

cout = sum((i-g_i).^2);
fprintf('cout = %d\n\n',cout);
end
```

Figure 8: Programme informatique d'optimisation.

Question 7 :

Le résultat du programme d'optimisation donne cette courbe pour des valeurs de L, R, Kc,Kv optimisés. Ainsi, par rapport à la courbe expérimentale, il y a un léger écart entre les deux courbes.

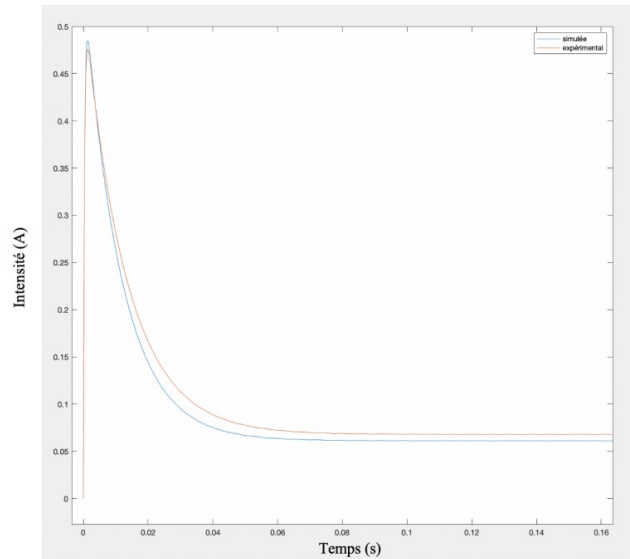


Figure 9: Résultats du programme d'optimisation.

3 Commande du moteur à courant continu

Question 8:

Voici le modèle classique d'un hacheur quatre quadrant qui est utilisée sur Matlab Simulink. (Les blocs ne sont pas en langage Matlab Simulink)

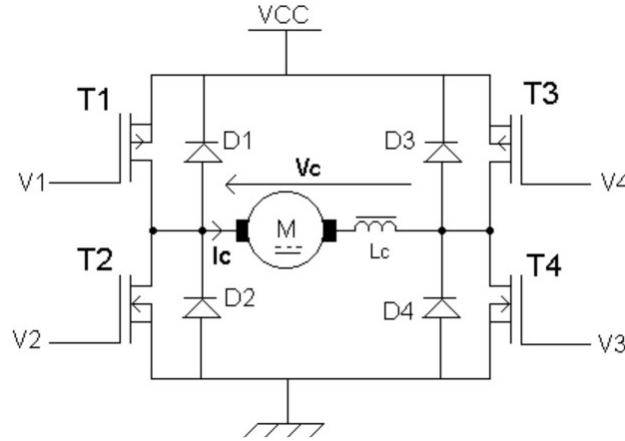


Figure 10: Modèle hacheur .

Suite à cela, nous allons envoyer des signaux de type créneaux avec des valeurs de rapport cyclique différentes (0; 0,5; 0,75).

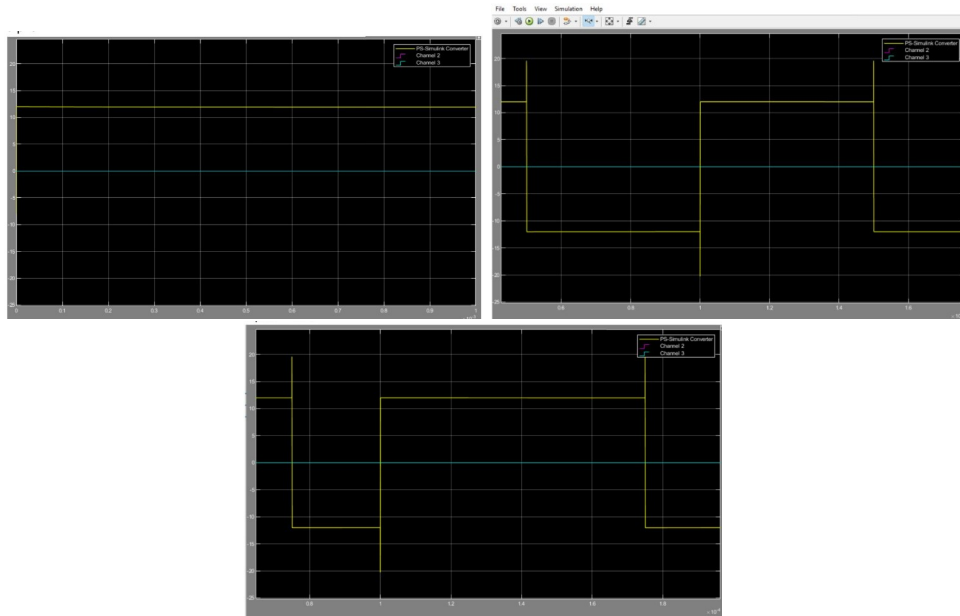


Figure 11: Hacheur pour des valeurs cycliques de 0; 0,5;0,75.

Or, d'après le cours, on a donc dans ce cas :

$$\langle V \rangle = V_{mot} = \frac{1}{T} \int_0^T v(t) dt = \frac{V(2t-T)}{T} = V_{al}(2\alpha - 1)$$

Question 9 :

Pour cette question nous allons établir que la puissance en entrée est égal à la puissance de sortie et on a, en utilisant la relation précédente :

$$V_{alimentation} I_{alimentation} = V_{moteur} I_{moteur}$$

$$V_{alimentation} I_{alimentation} = V_{alimentation} (2\alpha - 1) I_{moteur}$$

$$I_{alimentation} = I_{mot} (2\alpha - 1)$$

Question 10 :

Pour réaliser un modèle linéaire équivalent nous allons utiliser des blocs de Simulink pour simuler les relations utilisées précédentes. Pour cela on va sommer (à l'aide du sommeur) puis multiplier le signal pour avoir la relation trouvée précédemment :

$$V_{moteur} = V_{al} (2\alpha - 1); I_{alimentation} = I_{mot} (2\alpha - 1)$$

Ainsi, voici le résultat du modèle linéaire de l'hacheur :

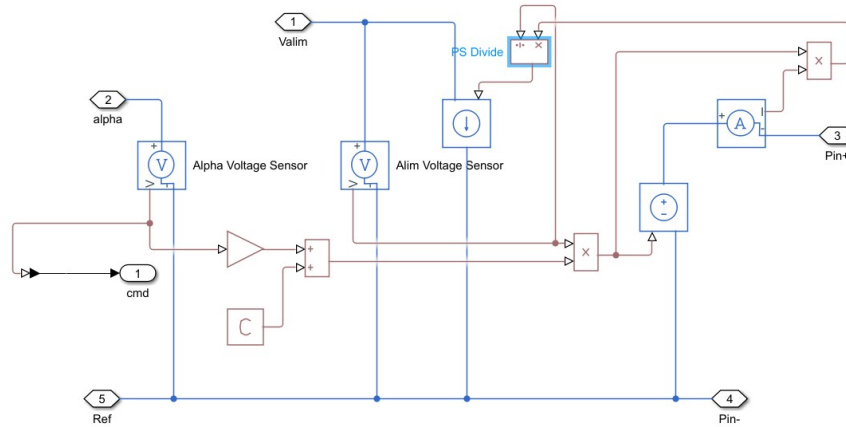


Figure 12: Nouveaux modèle de l'hacheur.

Question 11 :

Dans cette question, on nous demande de rajouter un correcteur proportionnel et intégrale. Ainsi, en utilisant la bibliothèque de Simulink le schéma revient, en mettant uniquement un intégrateur :

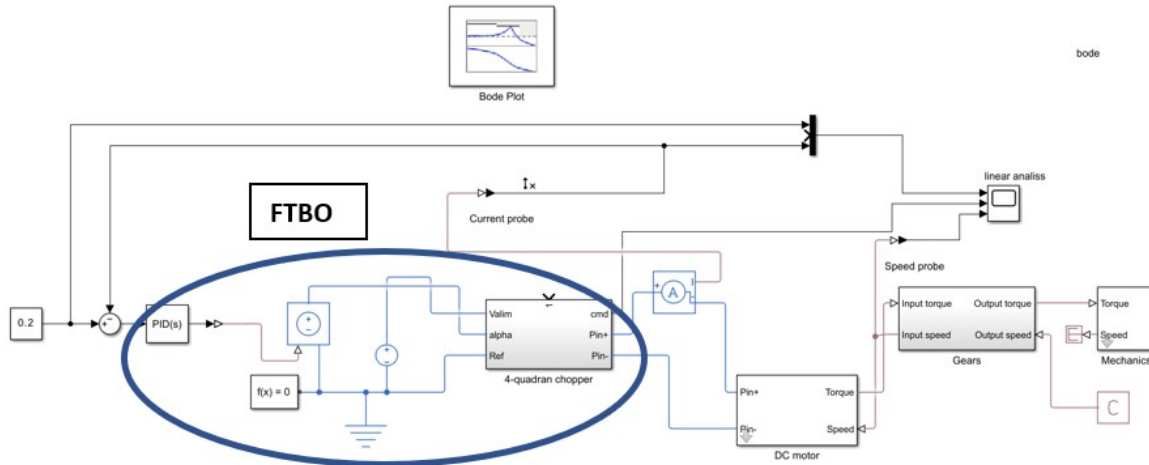


Figure 13: FTBO.

Question 12 :

Le système mise en évidence sur la boucle ouverte a été précisé dans la figure précédente. La FTBO peut être utilisée en utilisant le module "Linear Analysis Points/Open-Loop Input" et "Linear Analysis Points/Open-Loop Output". La FTBO est donc constitué du correcteur intégrale, du modèle de l'hacheur linéaire.

Question 13 :

On sait que le système est constitué d'un intégrateur qui est sous une forme :

$$PI(p) = K(1 + \frac{1}{p})$$

Ainsi, PI(p) additionnée à un hacheur de type linéaire serait donc un système d'ordre 1 et de classe 0.

Question 14 :

On peut aussi relever le diagramme de Bode de la FTBO et observer une pente de -20dB/décade et une phase de -90° et valider ainsi l'ordre 1 et la classe 0.

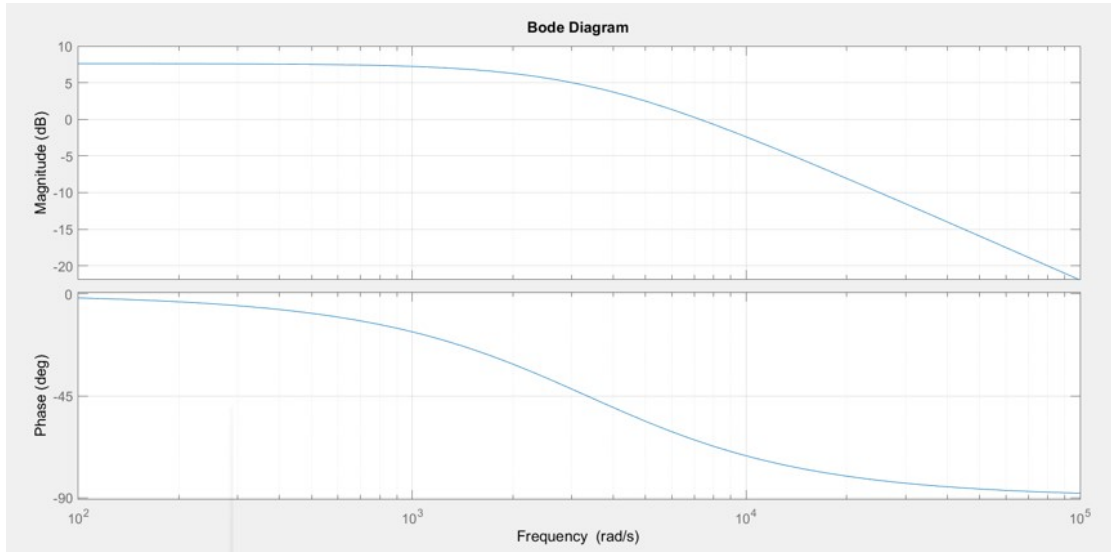


Figure 14: Diagramme de Bode de la FTBO.

Question 15 :

Dans l'onglet "Plots and Results" avec "Result Viewer" on arrive à prendre la FTBO de ce type :

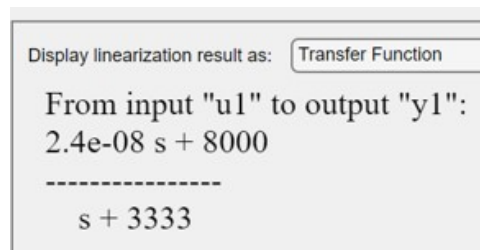


Figure 15: Nouveaux modèle de l'hacheur.

Cela est bien cohérent avec les résultats précédent, il s'agit, en effet, d'un passe bas d'ordre 1 et de classe 0 attendu avec $\omega_0 = 3333 \text{ rad/s}$

Question 16 :

Pour déterminer les paramètres du PI qui permettent un dépassement de 0,05 avec un temps de 200 microsecondes, on modifie qualitativement les paramètres pour s'approcher d'une valeur attendue. Le dépassement est de 0,052 et $t_5 = 240 \mu s$ pour $P = 6$; $I = 48200 \text{ rad/s}$

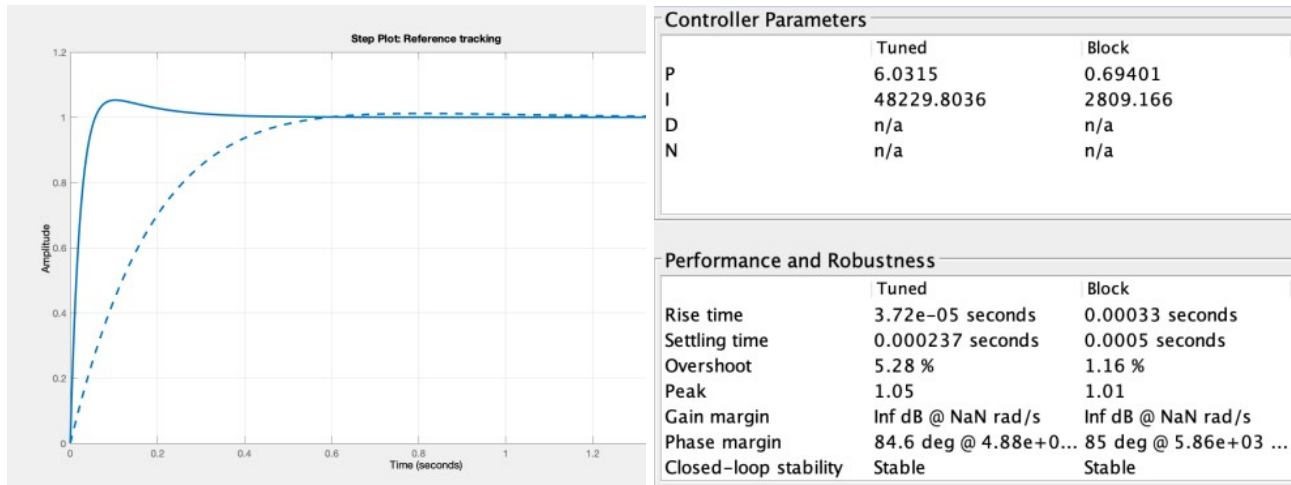
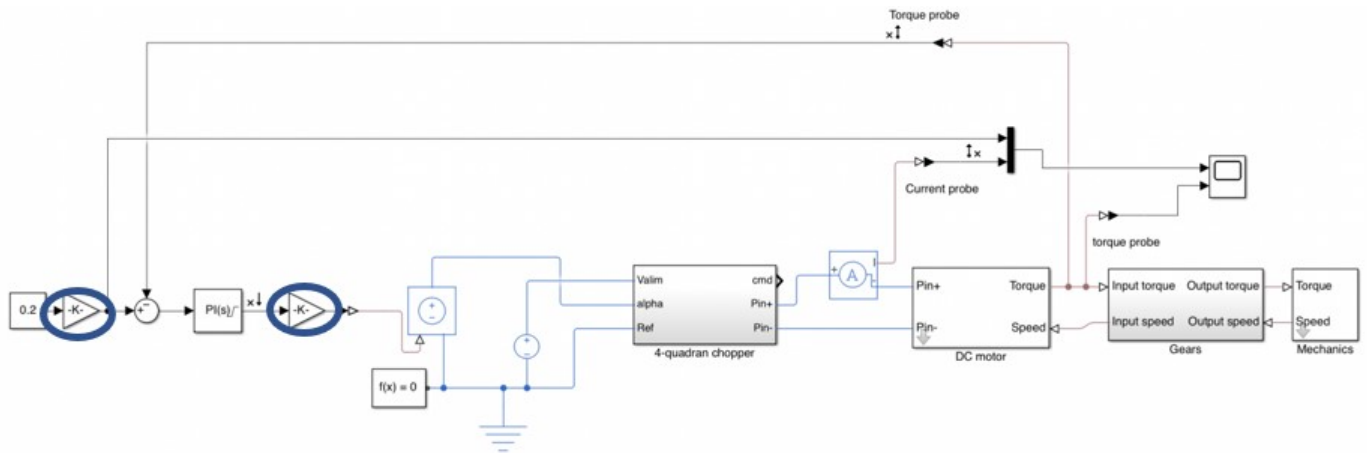


Figure 16: Valeurs du PI et système en sortie.

Question 17 :

Pour piloter le modèle en couple il faut pouvoir modifier une entrée en couple pour comparer ensuite les grandeurs. Voici donc le modèle pour piloter le couple (et non le courant). On peut alors observer :



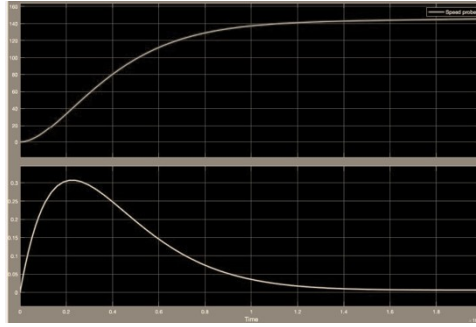


Figure 17: Couple en fonction des positions.

4 Modélisation mécanique du robot

Question 19:

Pour réaliser le trapèze de vitesse, avec une commande, on effectue la forme suivante dans le signal Builder. Suite à cela, on supprime le bloc intégrateur pour que le trapeze de vitesse soit invariant pour une certaine vitesse, accélération et position.

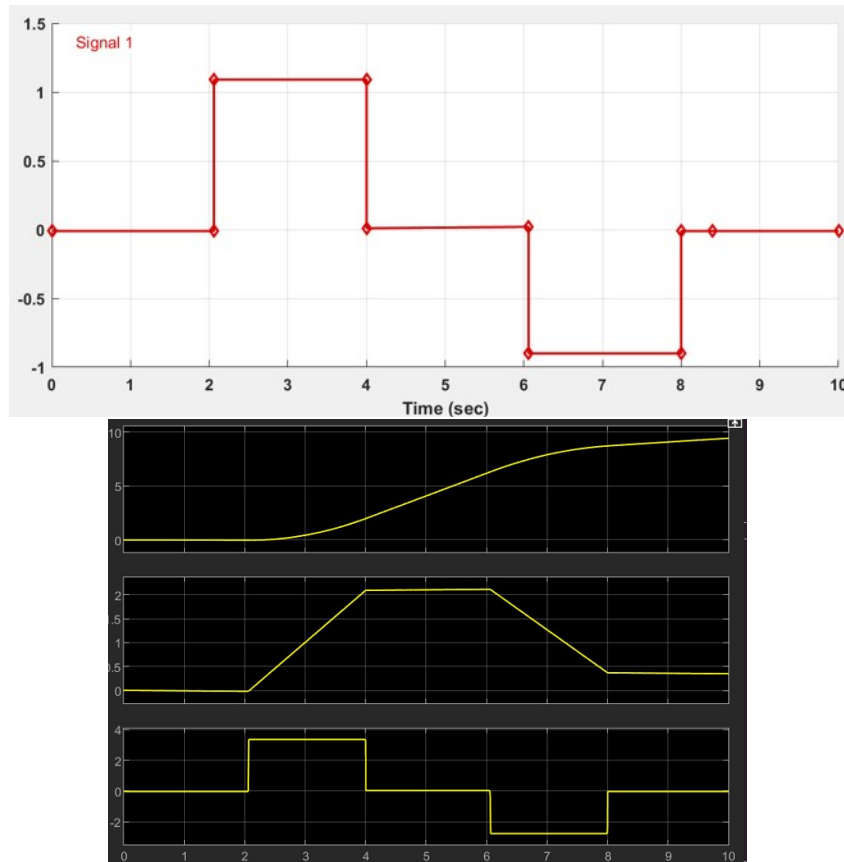


Figure 18: Trapèze des vitesses et accélérations .

Question 20 :

On réalise ainsi trois simulation pour les position de $0^\circ, 90^\circ, 180^\circ$:

- Pour un angle de 180° , $C_{\max} = 1 \text{ N.m}$
- Pour un angle de 90° , $C_{\max} = 1.6 \text{ N.m}$
- Pour un angle de 0° , $C_{\max} = 2.2 \text{ N.m}$

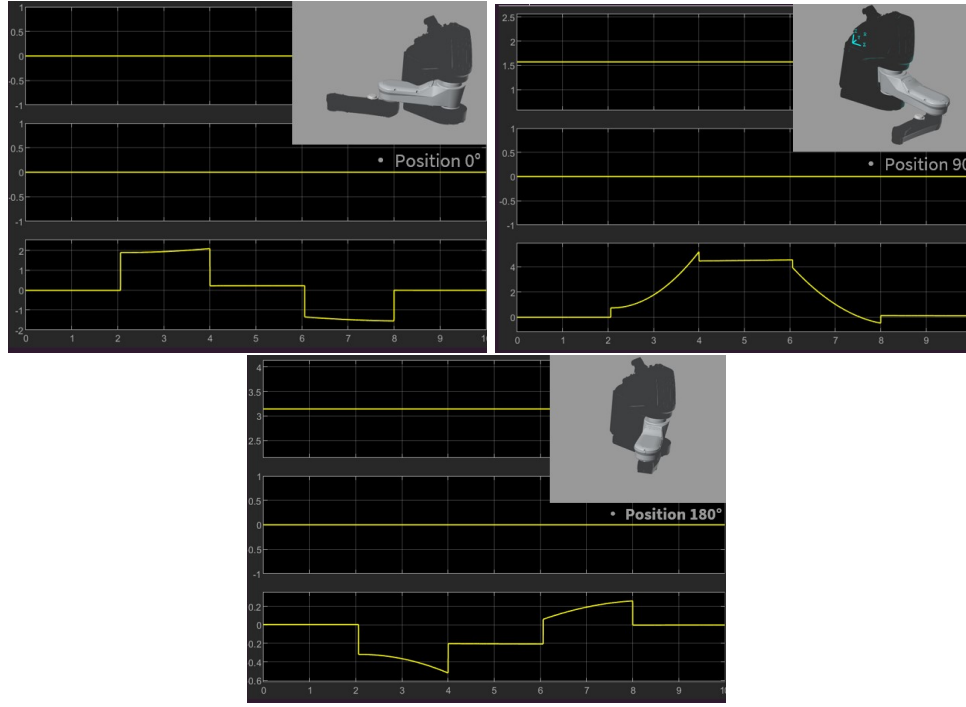


Figure 19: Couple en fonction des positions.

Question 21 :

La difference de couple peut s'expliquer par la différence d'inertie de la pièce. En effet, en fonction de la position de la pièce on aura une force d'inertie (de coriolis) qui fera changer la couple. Voici, la relation qui détermine cette force de coriolis :

$$\vec{F}_C = -2m\Omega(t)(\vec{e}_{axe} \wedge \vec{v})$$

Figure 20: Couple en fonction des positions.

5 Commande en vitesse du robot

Question 22 :

Tout d'abord, il faut relier les sorties en vitesse et en position aux modèle mécanique. Après, on utilise des blocs "scope" pour relever la position et la vitesse.

Cependant, pour réaliser une commande de vitesse nous devons placer des sommateurs. Pour établir les relations entre les sommateurs, on utilise les relations :

$$V_{mot} = Val(2\alpha - 1)$$

$$E(t) = Kv * \Omega$$

Donc on a :

$$\Omega = Val(2\alpha - 1)/Kv$$

Le modèle " speed control" est complété de la manière suivante :

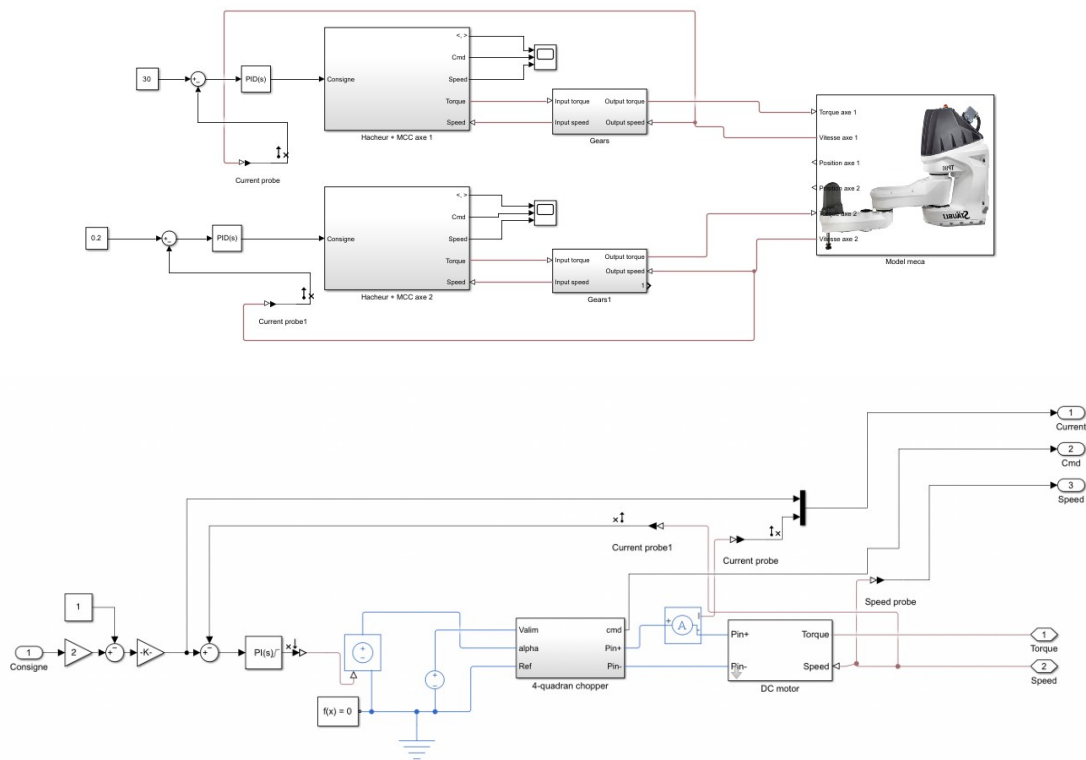


Figure 21: Nouveaux Modèle speed control.

Question 23 :

Avec ce modèle, on lance la simulation sur la commande en vitesse et on obtient ceci. On peut observer que le système est d'ordre 7. Par ailleurs, voici donc la sortie en vitesse du système :

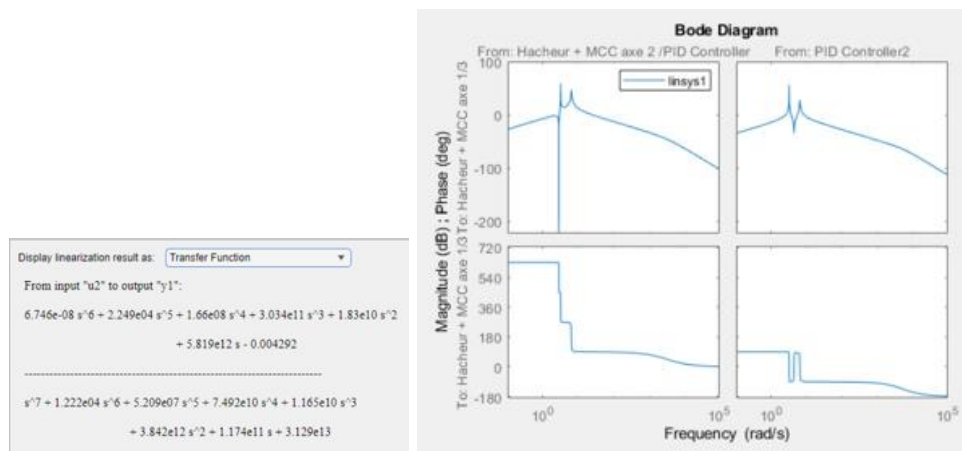


Figure 22: Fonction de transfert totale.

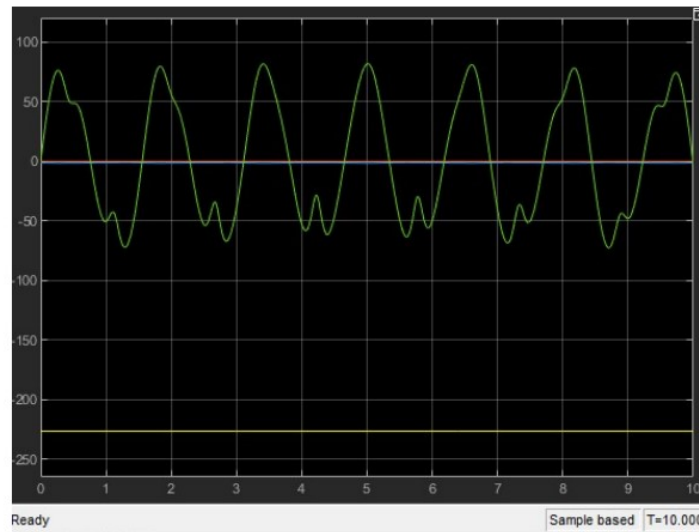


Figure 23: Sortie en vitesse du nouveau système.

6 Commande en position du robot

Question 26 :

En lançant la simulations sur la commande en deux positions, on obtient :

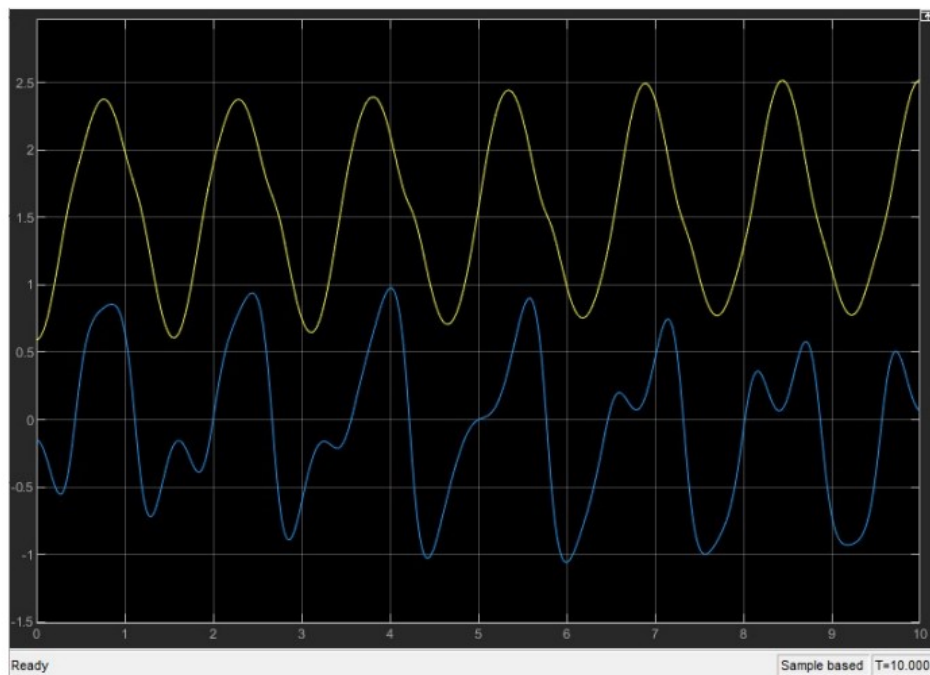


Figure 24: Sortie du système pour deux positions.

7 Conclusion

Pour conclure, ce TP d'outils numériques m'a permis de me familiariser avec les logiciels Matlab ; Simulink et le logiciel de traitement Latex. Des logiciels que je n'avais pas utilisés auparavant. Concernant, l'étude du robot il a été intéressant de proposer différents modèles sur les hacheurs et différentes modélisation pour la commande en position et en vitesse du moteur à courant continu. De même, cela m'a permis d'observer les influences mécaniques sur le système ainsi que les types de correcteurs adaptés pour en augmenter ses performances.