

Compte Rendu - TP5

Commande numérique : réponse pile et commande optimale

Janvier 2022

1 Introduction

L'objectif du TP est de concevoir deux commandes numériques pour contrôler un moteur électrique : commande pile et commande LQ.

Le moteur est modélisé par un premier ordre :

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{K_m}{1 + \tau s}$$

On utilise Matlab pour la commande. La mise en oeuvre numérique de la commande en temps réel est faite dans Matlab grâce à la boîte à outils "Simulink Desktop RealTime".

2 Régulation numérique avec correcteur proportionnel

On met en place une régulation échantillonnée comme ci-dessous :

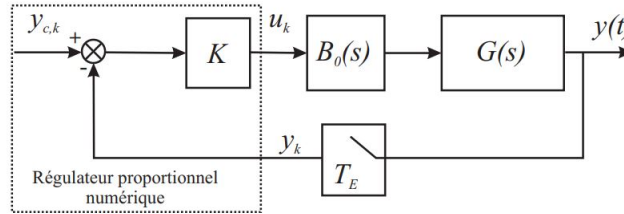


FIGURE 17 – Principe de bouclage discret

Calcul de la fonction de transfert échantillonnée du moteur lorsque celui-ci est précédé d'un convertisseur numérique analogique fonctionnant en bloqueur d'ordre zéro $B_o(p)$:

$$G(z) = K_m \frac{1 - z_0}{z - z_0}$$

avec $z_0 = \exp\left(-\frac{T_e}{\tau}\right)$

Par la suite, nous avons calculé la fonction de transfert en boucle fermée pour un régulateur proportionnel K , en fonction de K , K_m , T_E et τ .

$$H(z) = \frac{KG(z)}{1 + KG(z)}$$

$$H(z) = \frac{KK_m(1 - z_0)}{z - z_0 + KK_m(1 - z_0)}$$

2.1 Identification du procédé

Dans un premier temps nous cherchons à identifier les paramètres du moteur en étudiant sa réponse indicielle. Pour cela, sous Matlab, nous avons créé un modèle qui gère la commande. On envoie en entrée un échelon à 4 connecté au bloc CNA.

Nous obtenons en sortie la réponse du moteur avec un pas de calcul $T_e = 0.01s$ (ANNEXE 1).

On identifie ainsi le gain K_m du moteur; le régime permanent indique qu'en sortie le moteur atteint $K_m E = 6,9$ d'où $K_m = \frac{6,9}{4} = 1,73$. On peut de même identifier le temps de réponse du système en déterminant le $\tau_{63\%}$. Cette valeur est atteinte au bout de $t = \tau = 0,595s$.

Nous avons donc les paramètres suivants pour notre moteur :

$$K_m = 1,73$$

$$\tau = 0,595s$$

2.2 Auto-oscillations

2.2.1 Klim

Recherche des conditions de stabilité limite c'est-à-dire calcul de la surface de stabilité critique $K = f(T_e, \tau, K_m)$:

On cherche donc à résoudre :

$$-1 < |z| < 1 \Leftrightarrow -1 < |z_0 - K K_m (1 - z_0)| < 1$$

On s'intéresse à la partie gauche de l'inéquation:

$$K < \frac{1 + z_0}{K_m(1 - z_0)}$$

En gardant les paramètres du moteur trouvés précédemment, nous avons tracé la courbe $K = f(T_e, \tau, K_m)$ (ANNEXE 2).

On trouve ainsi pour $T_e = 0,35s$ un $K_{lim} = 2,022$ qui correspond donc à la limite de stabilité. On observe bien ce comportement en simulation (ANNEXE 3 et 4).

Le comportement en réel est bien celui escompté; le moteur se met à tourner, s'arrête, puis tourne à nouveau et ainsi de suite indéfiniment.

2.2.2 Erreur statique

L'erreur en régime statique est l'erreur qui résulterait de l'application d'une consigne constante (échelon). Elle est définie telle que :

$$\epsilon = \lim_{n \rightarrow \infty} \epsilon(n) = \lim_{z \rightarrow 1} (z - 1)\epsilon(z)$$

$$\epsilon = \lim_{z \rightarrow 1} z \frac{1}{1 + K K_m \frac{1 - z_0}{z - z_0}}$$

$$\epsilon = \frac{1}{1 + K K_m}$$

Nous vérifions en pratique la validité de l'expression de l'erreur statique. Le gain est fixé tel que $K = 1$ et on garde les paramètres τ, T_e et K_m inchangés.

En théorie, l'erreur statique vaut donc :

$$\epsilon = \frac{1}{1 + K K_m} = \frac{1}{1 + 1,73} = 0,3663$$

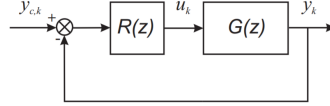
Comme l'entrée est un step de 4V, cela implique qu'on devrait trouver une réponse d'environ $4 - 0,3663 \times 4 = 2,535V$.

A l'aide du modèle Simulink en ANNEXE 5, on observe en sortie, la valeur lorsque le système est en régime permanent est de 2,437V, ce qui comparé aux 4V d'entrée indique une erreur de $4 - 2,437 = 1,56V$. D'où une erreur de $\epsilon = \frac{1,56}{4} = 0,39$.

On trouve quasiment la même erreur en statique que ça soit en simulation ou en réel.

2.3 Régulation numérique par commande à réponse pile

On désire réaliser une commande bouclée d'un procédé de fonction de transfert échantillonnée $G(z)$ avec un correcteur discret $R(z)$ comme ci dessous.



On a le système bouclé un modèle par fonction de transfert :

$$\frac{Y(z)}{Y_c(z)} = \frac{K_{damp}}{z^{d+1} - (1 - K_{damp})}$$

avec d =retard du procédé=0, $K_{damp} \in]0; 1] = gain_{amortissement}$

$$H(z) = \frac{K_{damp}}{z^{d+1} - (1 - K_{damp})} = \frac{R(z)G(z)}{1 + R(z)G(z)}$$

$$R(z)G(z) = K_{damp}(1 + R(z)G(z)) \frac{1}{z^{d+1} - (1 - K_{damp})}$$

$$R(z) = \frac{K_{damp}}{G(z)(z^{d+1} - 1)}$$

$$R(z) = \frac{K_{damp}}{K_m \frac{1-z_o}{z-z_o}(z-1)} = \frac{K_{damp}(z-z_o)}{K_m(1-z_o)(z-1)}$$

2.3.1 Simulation de la commande pile

Nous appliquons maintenant nos résultats théoriques au cas de la régulation de vitesse du moteur à l'aide d'une commande pile, pour $d = 0$ et $k_{damp} = [0.5; 0.8; 1]$.

En ANNEXE 6 et 7 se trouvent le modèle Simulink de la commande pile ainsi que la réponse en simulation.

On observe en simulation que la réponse du moteur est excellente et a un comportement quasiment similaire à la commande.

2.3.2 Mise en oeuvre réelle de la commande pile

On fait désormais varier le paramètre k_{damp} pour une mise en oeuvre réelle de la commande pile. On observe les résultats en ANNEXE 8, 9 et 10 (l'échelon est observé en rouge, la sortie de la fonction de transfert en jaune et la réponse du moteur en bleu).

On voit que la réponse du moteur s'améliore avec un temps d'établissement plus court lorsque l'on augmente k_{damp} . On observe par contre un dépassement plus important à la sortie de la fonction de transfert (pour $k_{damp} = 1$ par exemple).

2.4 Mise en œuvre d’une commande LQ à horizon infini

Pour réaliser la commande LQ, nous mettons le modèle dans sa représentation d’état qui est la suivante :

$$\begin{cases} \dot{x} = \frac{-1}{\tau}x + \frac{K_m}{\tau}u \\ y = x \end{cases}$$

$$K = R^{-1}B^T P$$

On pose pour cette section $Q = \rho \in R$ et $R = 1$. Le code MATLAB se situe en ANNEXE 11.

2.4.1 Simulation de la commande LQ sous Simulink

Le modèle pour la simulation de commande LQ en ANNEXE 12.

On observe alors pour différentes valeurs de ρ les réponses en ANNEXE 13, 14 et 15.

On remarque que, comme prévu, plus ρ est faible ($\ll 1$), plus on agit sur la minimisation de la commande. Les actionneurs ne sont alors pas trop sollicités. A l’inverse, plus ρ augmente ($\gg 1$), plus on cherche à filtrer rapidement la commande et ce au détriment de l’énergie nécessaire au fonctionnement du système.

2.4.2 Simulation réelle de la commande LQ

Nous pouvons ensuite comparer avec la mise en œuvre réelle de la commande LQ sur le système pour les mêmes valeurs de ρ en ANNEXE 16, 17 et 18.

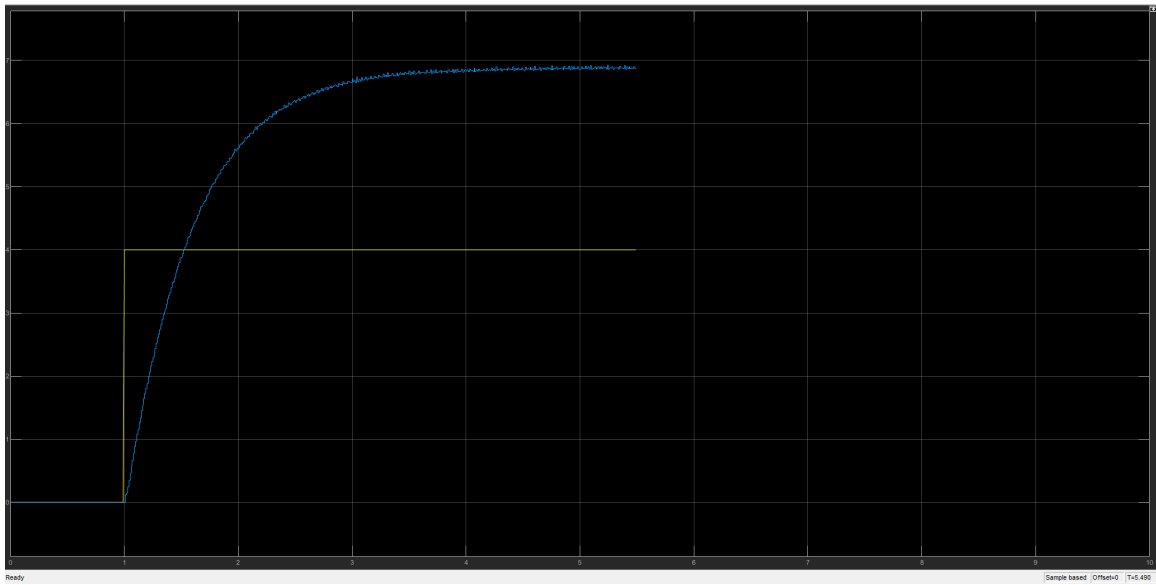
On remarque un comportement sensiblement similaire en réel pour $\rho \simeq 1$ mais pour un ρ plus élevé (par exemple 10 ici), la réponse est complètement saturée et va au-delà des valeurs limites de commande $[-10V, +10V]$.

3 Conclusion

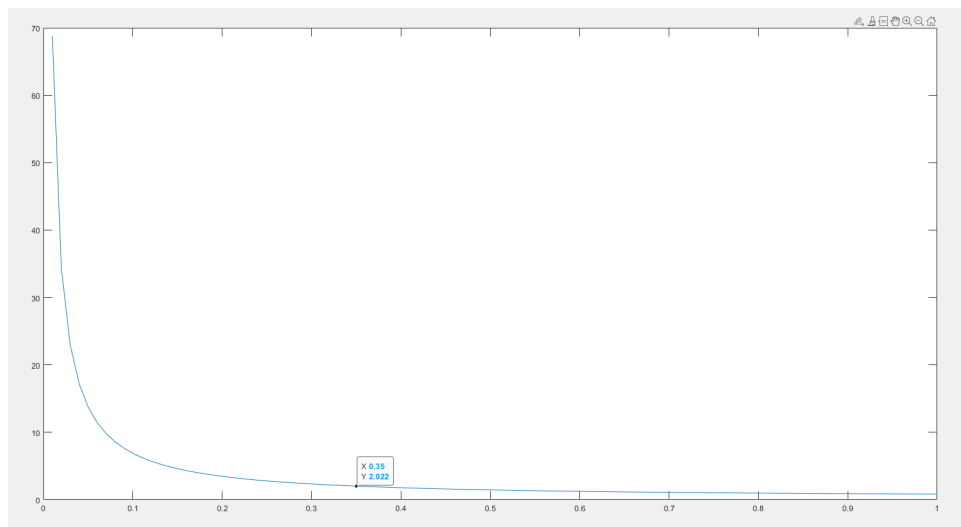
Grâce à ce TP nous avons pu réaliser l’implémentation de correcteurs sur un cas pratique, ici un moteur électrique. Nous avons pu revoir les systèmes discrets ainsi que les représentations d’état. Nous avons notamment remarqué des problèmes de saturation lors de la mise en place du régulateur LQ quand nous cherchions à affiner la correspondance réponse/commande.

4 ANNEXES

Identification du procédé

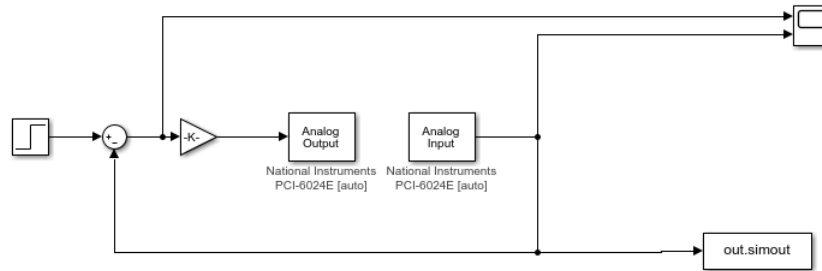


ANNEXE 1. Réponse moteur avec un pas de calcul $T_e=0.01s$

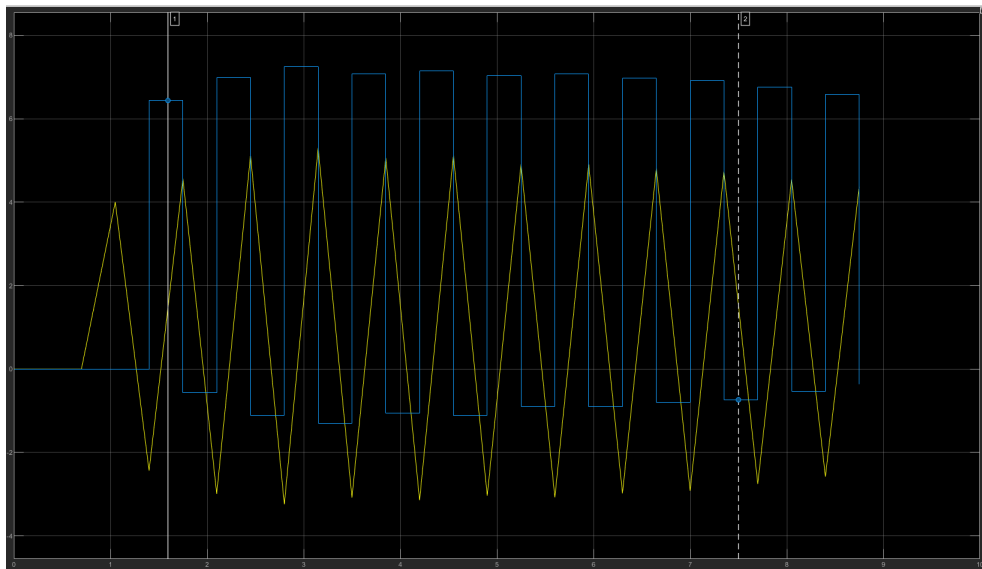


ANNEXE 2. Courbe $K = f(T_e, \tau, K_m)$

Auto-oscillations

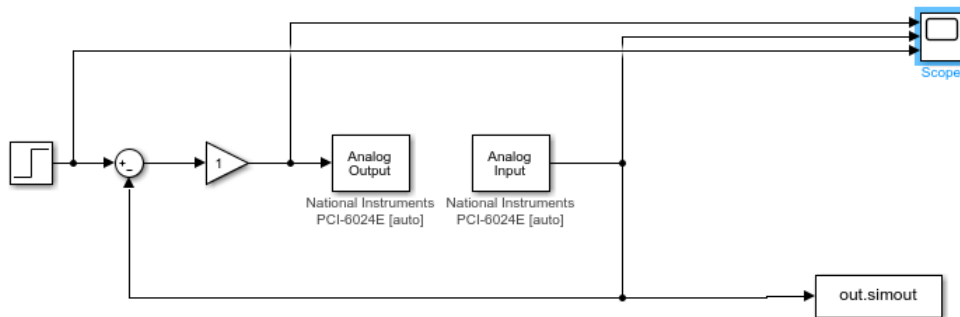


ANNEXE 3. Modèle Simulink du correcteur proportionnel



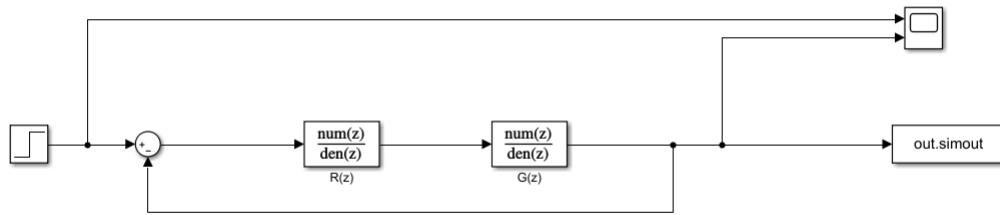
ANNEXE 4. Visualisation de la réponse du moteur au scope pour $K_{lim} = 2.022$

Erreur statique

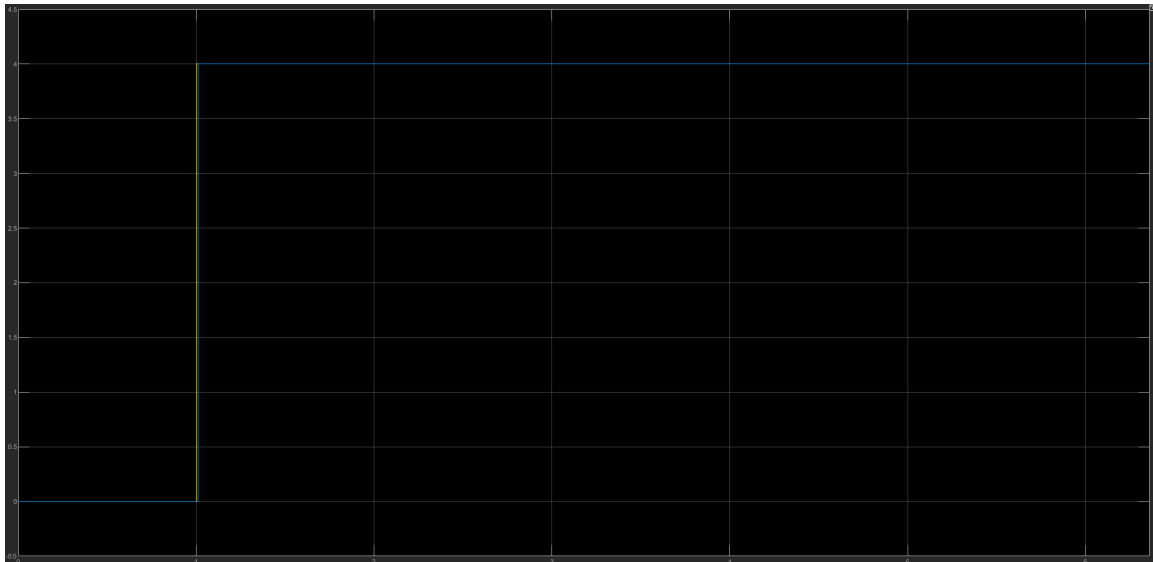


ANNEXE 5. Modèle Simulink correcteur proportionnel avec $K=1$

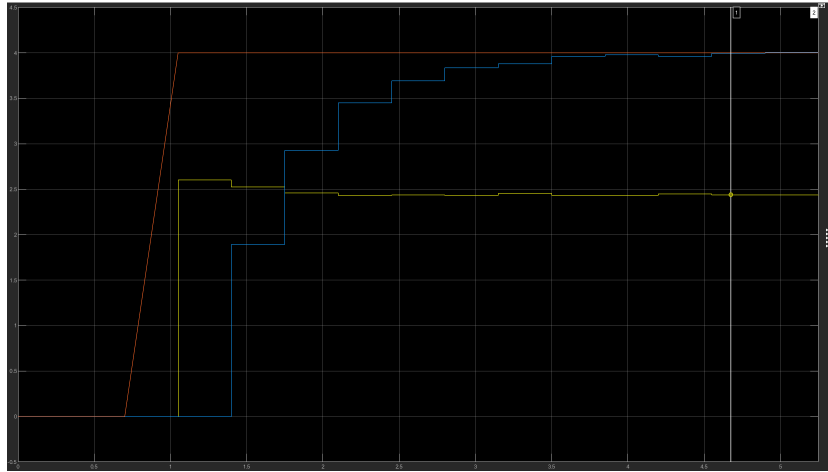
Commande pile



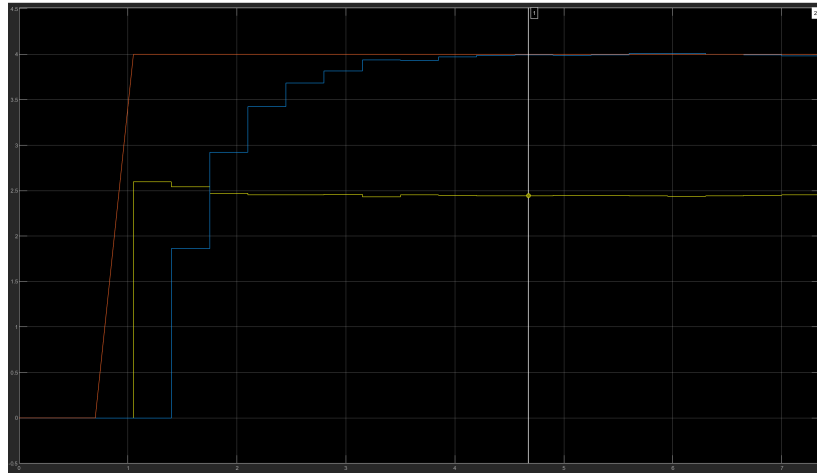
ANNEXE 6. Modèle Simulink de la commande pile



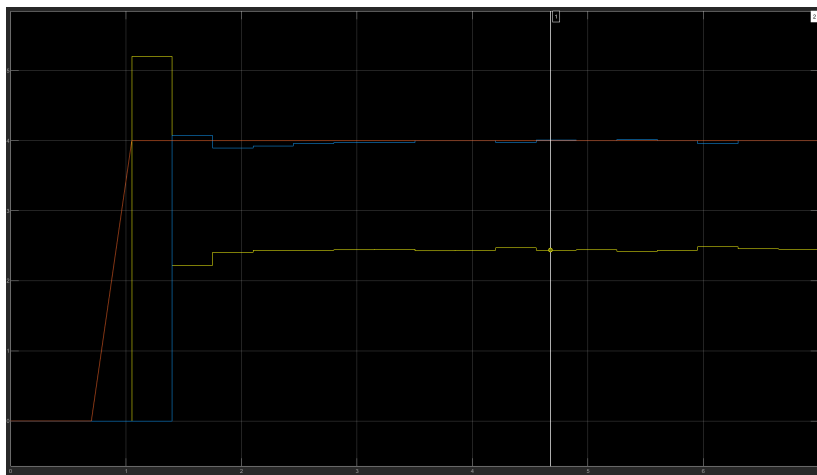
ANNEXE 7. Réponse à la simulation de la commande pile



ANNEXE 8. Réponse du moteur pour une commande pile : $kdamp = 0.5$



ANNEXE 9. Réponse du moteur pour une commande pile : $kdamp = 0.8$



ANNEXE 10. Réponse du moteur pour une commande pile : $kdamp = 1$

Commande LQ

```
Te = 0.35;
tau = 0.595;
Km = 1.73;

% COMMANDE LQ %

A = -1/tau;
B = Km/tau;
C = 1;
D = 0;

sys = ss(A,B,C,D);

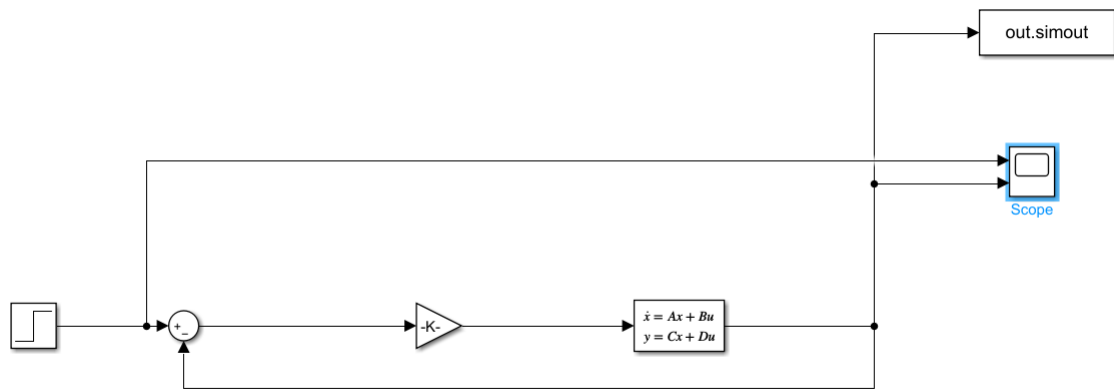
Q = 0.02;

%Faire varier Q pour voir les changements sur l'énergie/commande

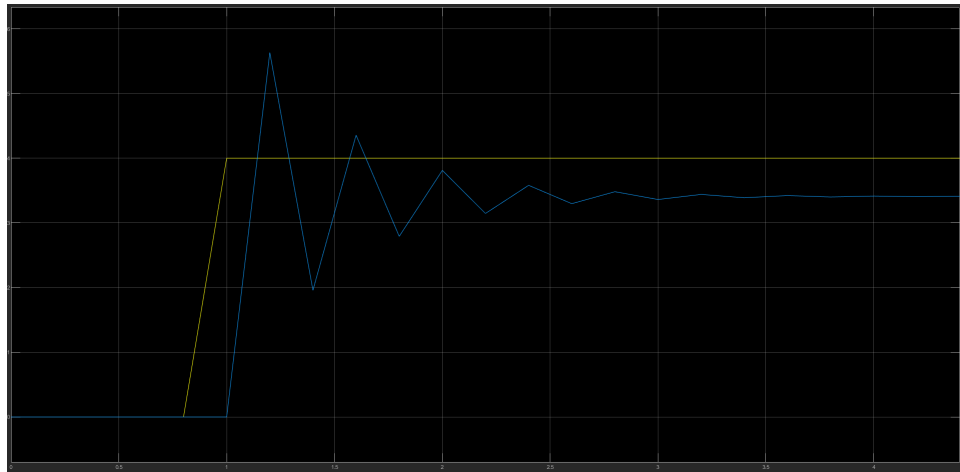
R = 1;
K =lqr(A,B,Q,R);

Abf = A-B*K;
sys_corrige = ss(Abf, B, C, D);
nyquist(sys_corrige);
```

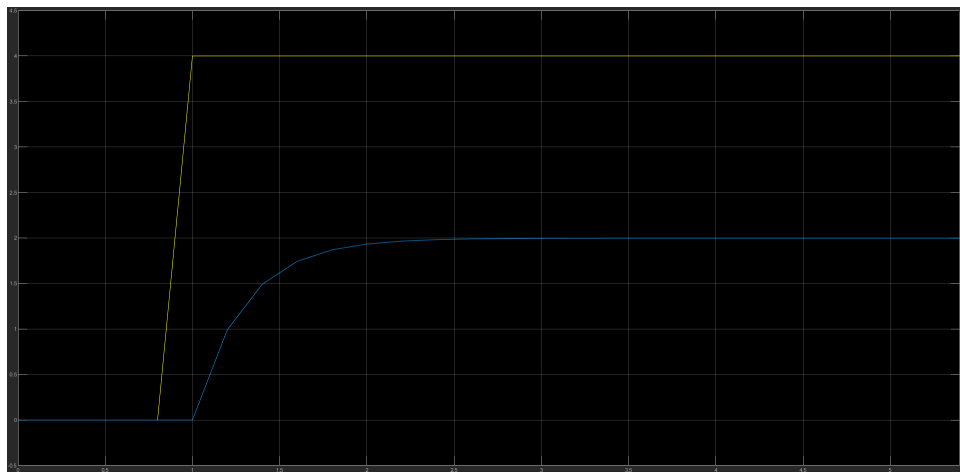
ANNEXE 11. Code Matlab pour la commande LQ



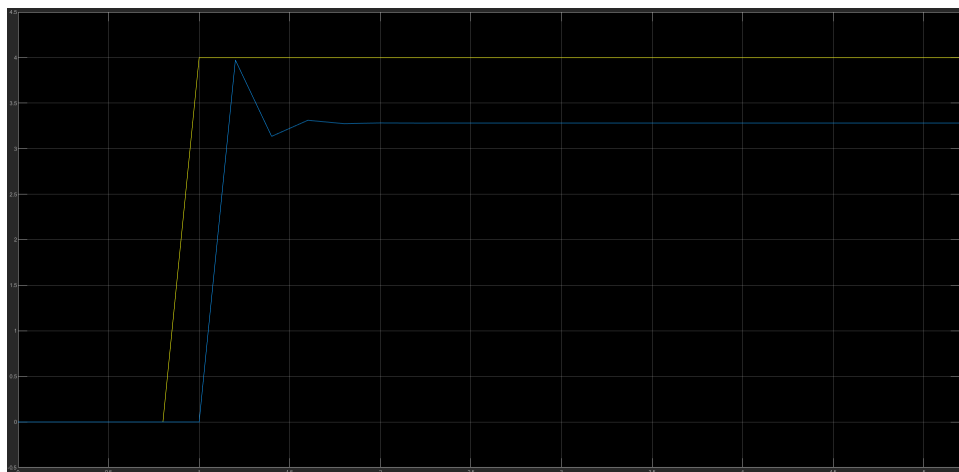
ANNEXE 12. Modèle pour la simulation de commande LQ



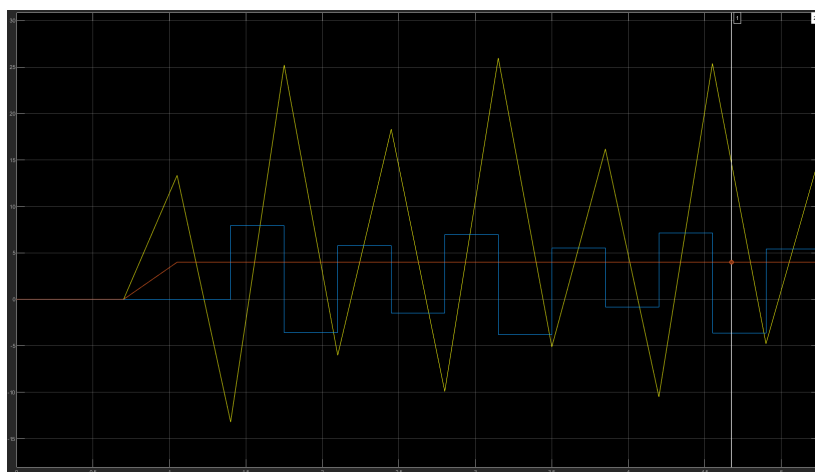
ANNEXE 13. Réponse de la simulation de la commande LQ pour $\rho = 0.1$



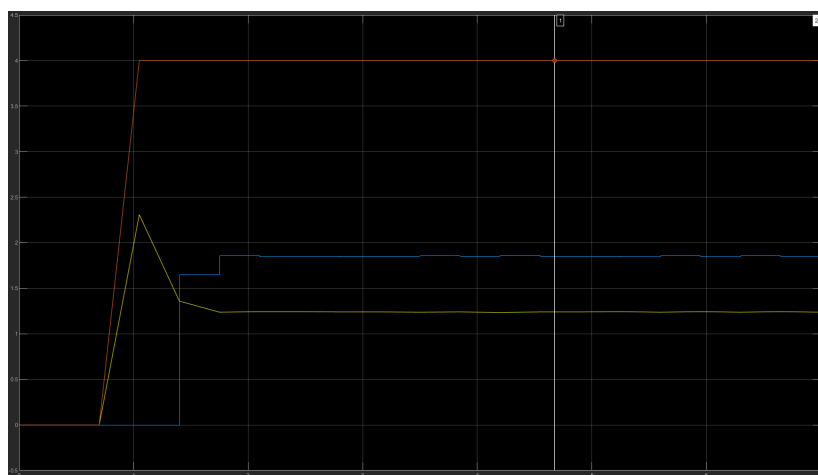
ANNEXE 14. Réponse de la simulation de la commande LQ pour $\rho = 1$



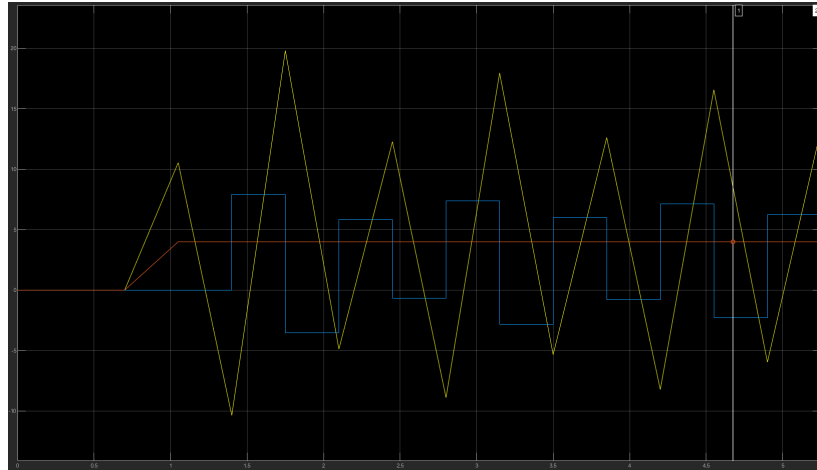
ANNEXE 15. Réponse de la simulation de la commande LQ pour $\rho = 10$



ANNEXE 16. Réponse du moteur à une commande LQ pour $\rho = 0.1$



ANNEXE 17. Réponse du moteur à une commande LQ pour $\rho = 1$



ANNEXE 18. Réponse du moteur à une commande LQ pour $\rho = 10$