

TP

Automatique

Rapport sur le TP5
Commande numérique : réponse pile & Commande
optimale

26 janvier 2021

Étudiants :

- Piques Nicolas
- Pitault Léa

Introduction

Tout au long de ce TP, nous allons asservir un moteur électrique selon différentes méthodes. Dans un premier temps, nous étudierons un asservissement numérique avec un gain proportionnel. Ensuite, nous mettrons en œuvre la commande à réponse pile. Enfin, nous appliquerons la commande linéaire quadratique LQ. Ces méthodes seront mises en place grâce au logiciel Matlab et Simulink Desktop RealTime.

1) Étude théorique

Notre fichier Matlab pour toute l'étude théorique se trouve en annexe ainsi que le schéma Simulink utilisé pour la simulation et pour la travail en réel sur le moteur.

a) Régulation numérique avec correcteur proportionnel

L'objectif ici est d'étudier un correcteur proportionnel à notre système et d'étudier l'impact de la période d'échantillonnage.

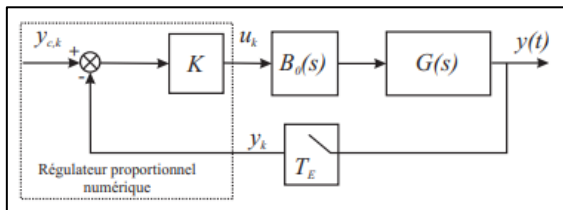


Figure 1 - Schéma-bloc du système avec gain proportionnel

- Fonction de transfert échantillonnée du moteur $G(z) = Z[B_0(p)G(p)]$:

$$G(z) = K_m \frac{(1 - e^{-\frac{TE}{\tau}})}{(z - e^{-\frac{TE}{\tau}})}$$

- Fonction de transfert en boucle fermée :

$$FTBF = \frac{FTBO}{1 + FTBO} = \frac{K \cdot G(z)}{1 + K \cdot G(z)} = \frac{K \cdot K_m (1 - e^{-\frac{TE}{\tau}})}{z - e^{-\frac{TE}{\tau}} + K \cdot K_m \cdot (1 - e^{-\frac{TE}{\tau}})}$$

- L'erreur en régime permanent d'après le théorème de la valeur finale est donnée par :

$$\text{erreur} = \lim_{z \rightarrow 1} (z - 1) \cdot \frac{z}{(z - 1)} \cdot \frac{1}{1 + FTBO} = \frac{1 - e^{-\frac{TE}{\tau}}}{(1 - e^{-\frac{TE}{\tau}})(1 + K \cdot K_m)} = \frac{1}{1 + K \cdot K_m}$$

- Nous choisissons d'appliquer le critère de Routh afin de trouver les conditions de stabilité limité et d'en déduire la surface de stabilité critique : $K = f(TE, \tau, K_m)$:

$$\text{Soit } z = \frac{1+w}{1-w}$$

$$\text{Nous avons comme équation caractéristique : } P(z) = z - e^{-\frac{TE}{\tau}} + K \cdot K_m \cdot (1 - e^{-\frac{TE}{\tau}})$$

$$\text{D'où : } P(w) = \frac{1+w}{1-w} - e^{-\frac{TE}{\tau}} + K \cdot K_m \cdot (1 - e^{-\frac{TE}{\tau}})$$

Finalement, le critère de Routh sur ce polynôme nous donne les conditions suivantes :

$$\frac{-1}{K_m} < K < \frac{1 + e^{-\frac{TE}{\tau}}}{(1 - e^{-\frac{TE}{\tau}}).K_m}$$

K étant positif, la seule limite qui nous intéresse est $K < \frac{1 + e^{-\frac{TE}{\tau}}}{(1 - e^{-\frac{TE}{\tau}}).K_m}$

b) Régulation numérique par commande à réponse pile

Nous souhaitons à présent étudier une commande à réponse pile. Cette commande a pour objectif d'obtenir une erreur nulle en régime permanent après un temps fini.

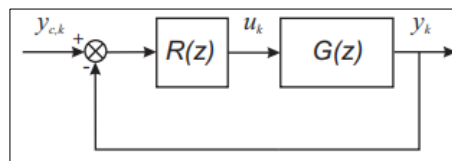


Figure 2 - Schéma-bloc du système avec commande à réponse pile

Nous souhaitons la fonction de transfert suivante : $\frac{Y(z)}{Y_c(z)} = \frac{k_{damp}}{z^{(d+1)} - (1 - k_{damp})}$ ce qui donne une erreur nulle en régime permanent.

$$\text{Soit : } G(z) = \frac{N(z)}{D(z)} = K_m \frac{(1 - e^{-\frac{TE}{\tau}})}{(z - e^{-\frac{TE}{\tau}})} \quad \text{et} \quad R(z) = \frac{N_R(z)}{D_R(z)}$$

$$\text{Ainsi, nous avons} \quad \frac{Y(z)}{Y_c(z)} = \frac{R(z).G(z)}{1 + R(z)G(z)} = \frac{N(z).N_R(z)}{D(z).D_R(z) + N(z).N_R(z)} = \frac{k_{damp}}{z^{(d+1)} - (1 - k_{damp})}$$

Nous pouvons donc déduire par identification :

$$\begin{cases} N(z).N_R(z) = k_{damp} \\ D(z).D_R(z) + N(z).N_R(z) = z^{(d+1)} - 1 + k_{damp} \end{cases} \Rightarrow \begin{cases} N_R(z) = \frac{k_{damp}}{N(z)} = \frac{k_{damp}}{K_m(1 - e^{-\frac{TE}{\tau}})} \\ D_R(z) = \frac{z^{(d+1)} - 1 + k_{damp} - N(z).N_R(z)}{D(z)} = \frac{z^{(d+1)} - 1}{z - e^{-\frac{TE}{\tau}}} \end{cases}$$

$$\text{Par conséquent, notre correcteur s'écrit sous la forme : } R(z) = \frac{k_{damp} \cdot (z - e^{-\frac{TE}{\tau}})}{K_m \left(1 - e^{-\frac{TE}{\tau}}\right) \cdot (z^{(d+1)} - 1)}$$

c) Contrôle par approche LQ

Le contrôle par approche LQ est une méthode par essai-erreur afin de déterminer le gain proportionnel K. On souhaite minimiser la fonction de coût propre au système et à la commande. Pour cela, on fait varier le scalaire p venant de Q = pI_n de sorte à trouver un compromis entre la minimisation de l'énergie du système et la minimisation de la commande.

Dans un premier temps, nous mettons notre modèle sous forme de représentation d'état.

$$\text{Nous avons : } \frac{Y(s)}{U(s)} = \frac{Km}{1 + \tau s} \Rightarrow y(t) + \tau \cdot \dot{y}(t) = Km \cdot u(t)$$

$$\Leftrightarrow \dot{y}(t) = \frac{Km \cdot u(t) - y(t)}{\tau}$$

$$\begin{cases} \dot{X} = AX + B \\ Y = CX + D \end{cases} \quad \text{avec} \quad \begin{cases} A = -\frac{1}{\tau} \\ B = \frac{Km}{\tau} \\ C = 1 \\ D = 0 \end{cases}$$

Figure 3 - Modèle d'état

L'asservissement se réalise selon le schéma bloc suivant :

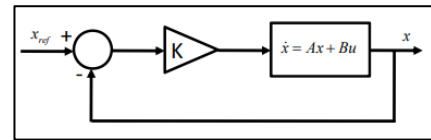


Figure 4 - Schéma-bloc avec contrôle par approche LQ

Plus p est grand et plus on diminue l'erreur statique.

Inversement, plus p est petit et plus on économise de l'énergie.

L'objectif est de jouer sur p pour trouver un équilibre entre ces deux paramètres.

2) Étude expérimentale

a) Identification du procédé

Nous allons commander un moteur électrique à l'aide de la boîte à outils "Simulink Desktop RealTime" de Matlab. Nous mettons en place notre système avec un échelon de 4V en entrée et nous obtenons la *figure 1* en annexe.

Nous en déduisons nos paramètres du système :

Km	2.04
τ	0.6s

b) Auto-oscillations

Les paramètres de notre moteur étant définis, nous allons étudier les limites de celui-ci. Pour se faire, nous traçons la courbe $K = f(T_E, \tau, K_m)$ et nous obtenons la *figure 2* en annexe.

Nous remarquons que plus le temps d'échantillonnage est long, plus, pour une valeur de K faible, le système rentre en auto-oscillations. Il sera donc intéressant de prendre un temps d'échantillonnage plutôt faible pour implémenter notre correcteur.

Par exemple, pour une période d'échantillonnage $T_E = 0.35s$, nous trouvons $K_{lim} = 1,72$ avec la formule de l'étude théorique. Avec ce K_{lim} notre système oscille. Sur les *figures 3 et 4* sont illustrées les réponses de notre système asservi. Nous remarquons en effet qu'en simulation et en réel nous avons bien des oscillations.

Il est également possible de calculer l'erreur statique pour $K=1$ avec toujours la même période d'échantillonnage. En théorie, la formule nous donne une *erreur* = 0,33. Ceci est vérifié en simulation et en réel sur les *figures 5 et 6*.

c) Mise en œuvre d'une commande à réponse pile

Nous appliquons notre commande par réponse pile sur le système d'après la théorie. Nous allons réguler la vitesse du moteur pour $d = 0$ et $k_{damp} = \{0.5 / 0.8 / 1\}$.

Nous remarquons sur les *figures 7, 8 et 9* que plus le k_{damp} est grand et plus la réponse pile est atteinte rapidement.

Nous avons malheureusement eu des problèmes avec la maquette pour tester ces résultats en réel. Nous devons donc nous contenter des résultats de simulation. Cependant, nous étions censés obtenir la même chose, c'est-à-dire une réponse pile atteinte plus rapidement lorsque k_{damp} se rapproche de 1. Notre système, lui, semble se rapprocher d'une auto-oscillation lorsque k_{damp} augmente.

d) Mise en œuvre d'une commande LQ à horizon infini

Au cours de cette expérimentation, nous allons faire varier p ce qui nous donnera une valeur de K , gain qui sera ensuite appliqué à notre système.

D'après les *figures 10, 11, 12 et 13*, on constate que plus p augmente, plus notre système se rapproche de la consigne mais, en contrepartie, il se déstabilise fortement. $p = 1$ semble être le meilleur compromis entre stabilité et erreur en régime permanent.

Conclusion

D'après nos différentes manipulations, il semble que la mise en œuvre de la commande à réponse pile soit la meilleure. Il ne nous a cependant pas été possible de confirmer cela en réel. La commande LQ reste cependant intéressante pour étudier l'impact que peut avoir un gain proportionnel mal choisi.

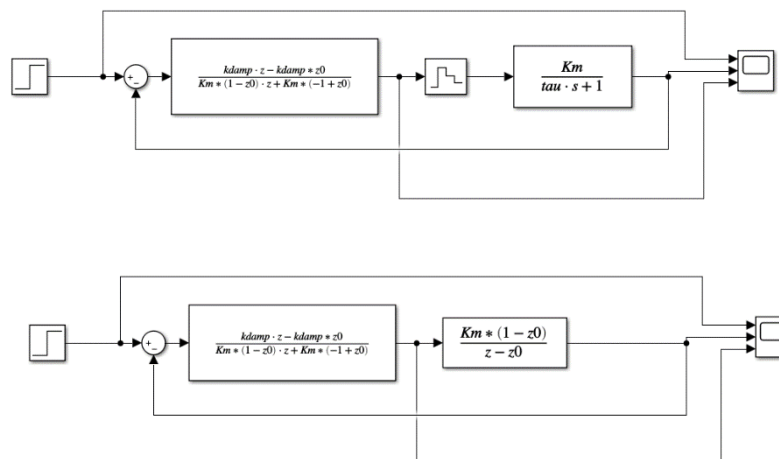
Annexes

```

1  clc;
2  %Temps d'echantillonnage
3  %Te=[0:0.1:100];
4  Km=2.04;
5  tau=0.6;
6
7  % for i = 1:1:100
8  %   Te(i)=0.01*i;
9  %   oscillations(i)=(1+exp(-Te(i)/tau))/(Km*(1-exp(-Te(i)/tau)));
10 %   end
11 % plot(Te,oscillations)
12
13 Te=0.35;
14 oscillilimite=(1+exp(-Te/tau))/(Km*(1-exp(-Te/tau)));
15 %K=1;
16 erreur=1/(1+K*Km);
17
18 kdamp=0.5;
19 z0=exp(-Te/tau);
20
21 %Calcul de K via LQ
22 A=-1/tau;
23 B=Km/tau;
24 C=1;
25 D=0;
26
27 Q=2.5;
28 R=1;
29 K=lqr(A,B,Q,R);

```

Code Matlab utilisé pour la partie théorique



Schémas Simulink utilisés pour la partie théorique en simulation

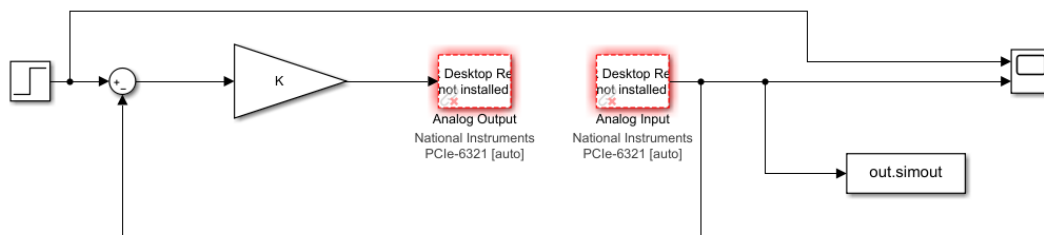


Schéma Simulink utilisé pour la partie réelle (STR)

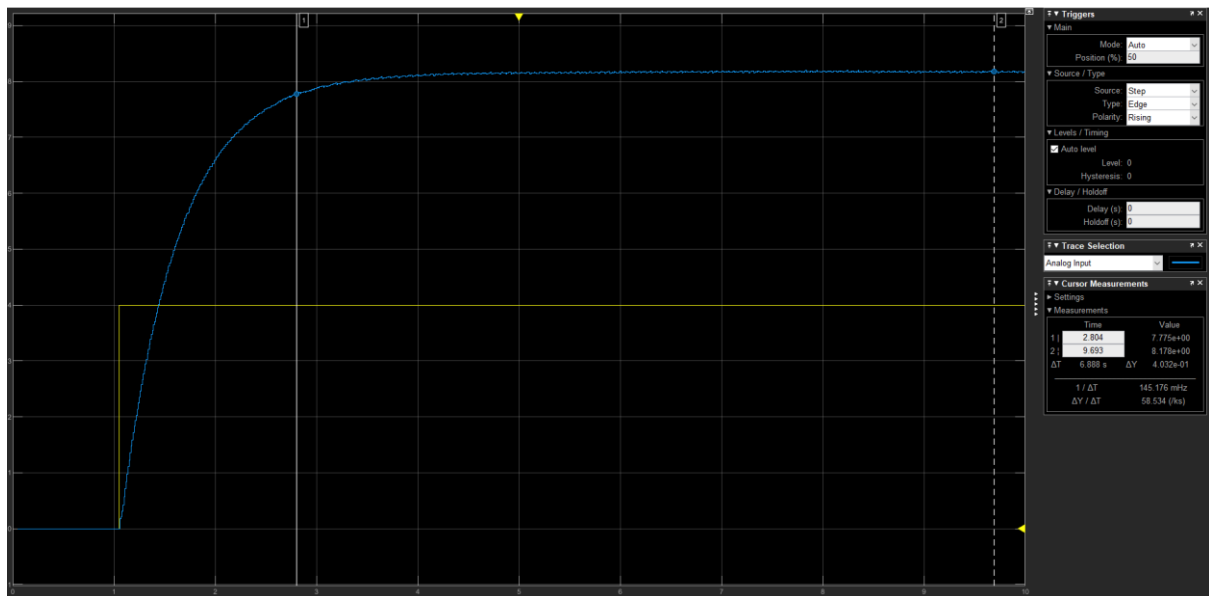


Figure 1 - Réponse du moteur à un échelon de 4V en entrée

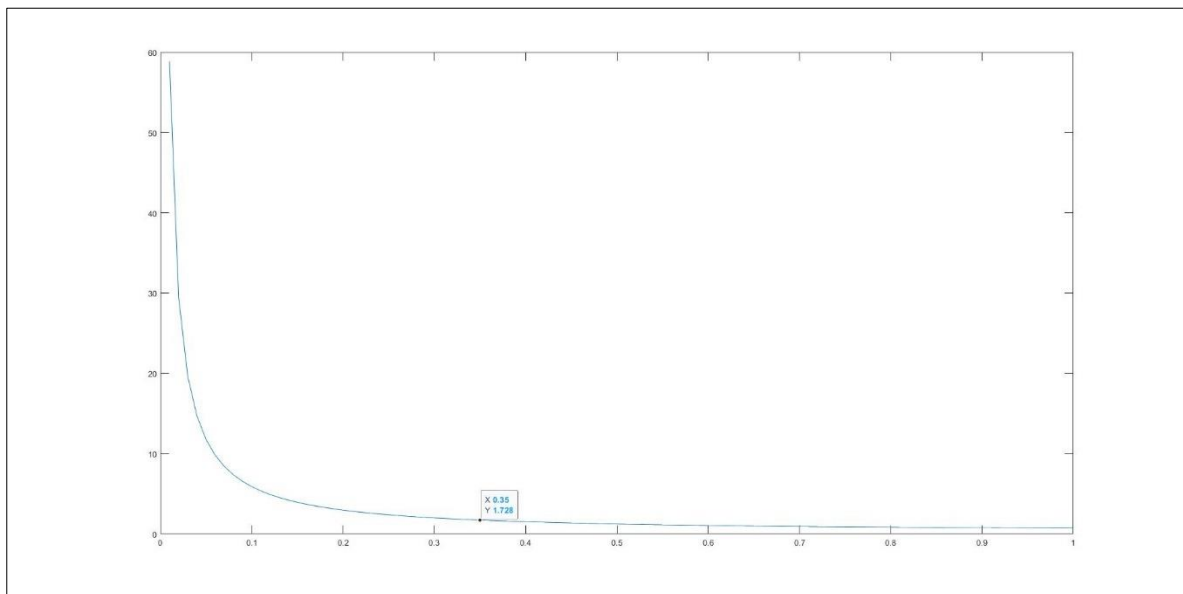


Figure 2 – Gain K_{lim} en fonction de T_e

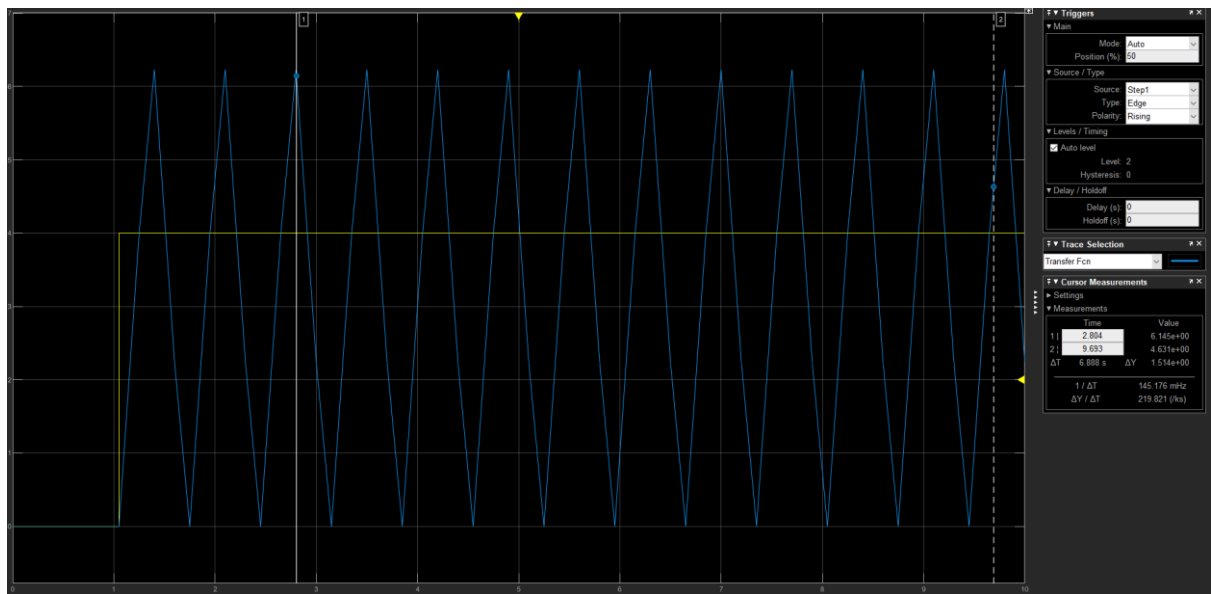


Figure 3 – Système oscillant avec le K_{lim} en simulation

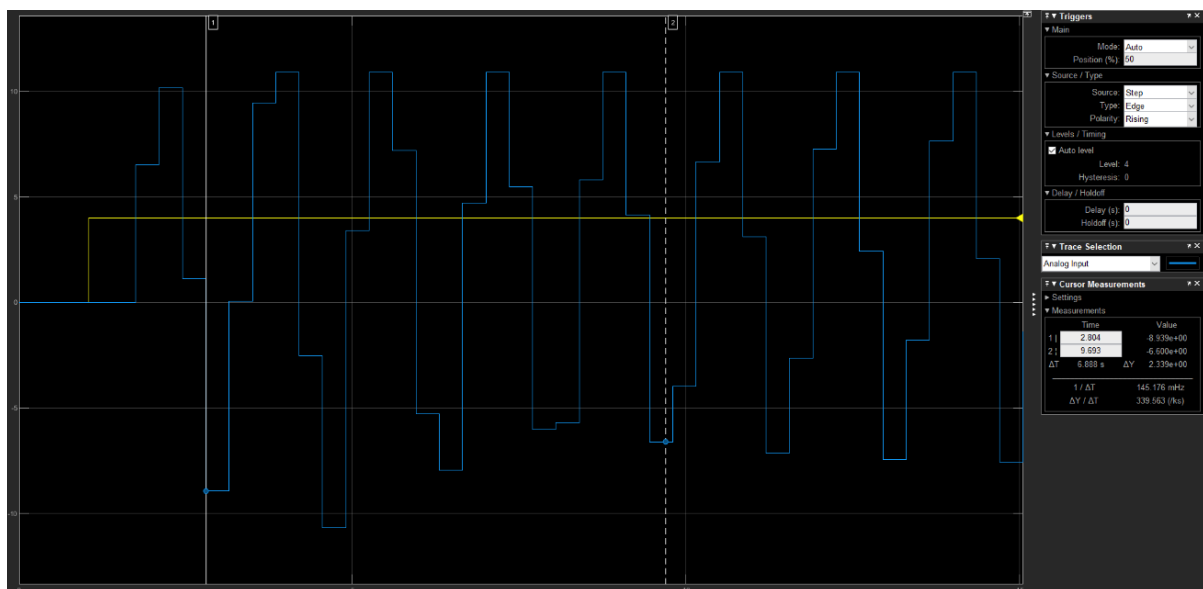


Figure 4 – Système oscillant avec le K_{lim} en réel

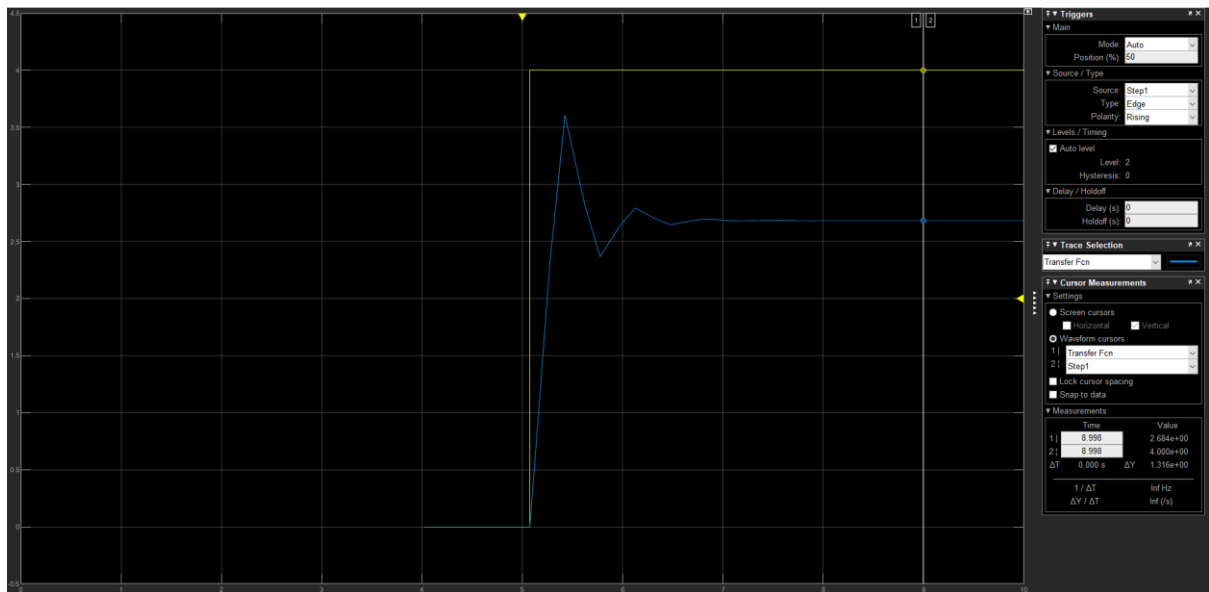


Figure 5 – Système avec $K=1$ permettant de déduire l'erreur statique en simulation

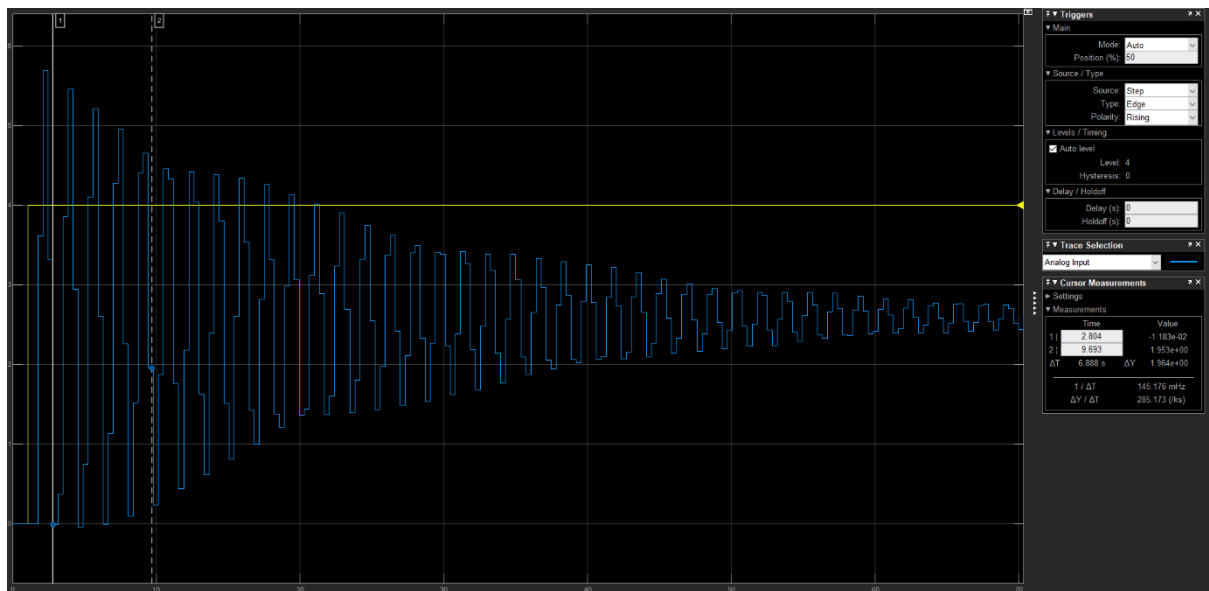


Figure 6 – Système avec $K=1$ permettant de déduire l'erreur statique en réel



Figure 7 – Réponse pile en simulation avec $K_{damp} = 0.5$



Figure 8 – Réponse pile en simulation avec $K_{damp} = 0.8$



Figure 9 – Réponse pile en simulation avec $K_{damp} = 1$

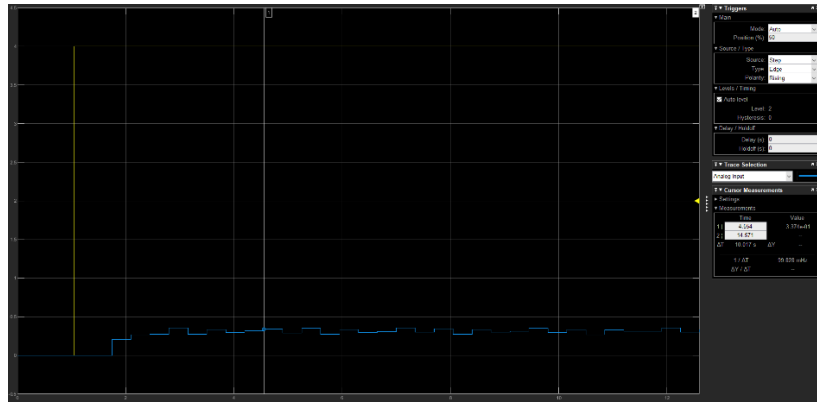


Figure 10 - Commande LQ ($p=0.1$) appliquée sur le moteur

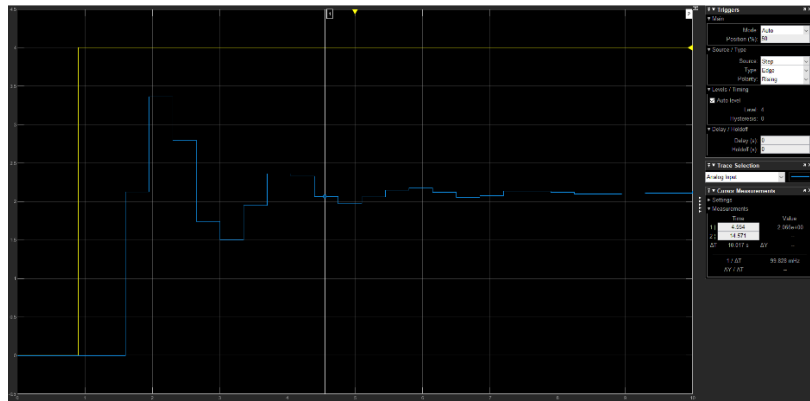


Figure 11 - Commande LQ ($p=1$) appliquée sur le moteur

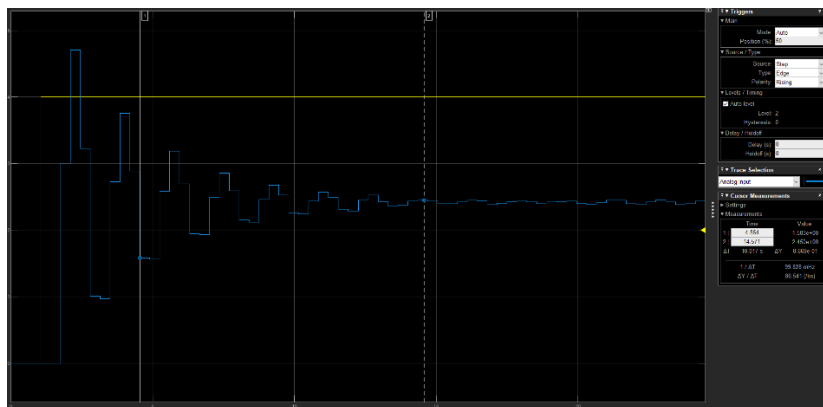


Figure12 - Commande LQ ($p=1.5$) appliquée sur le moteur

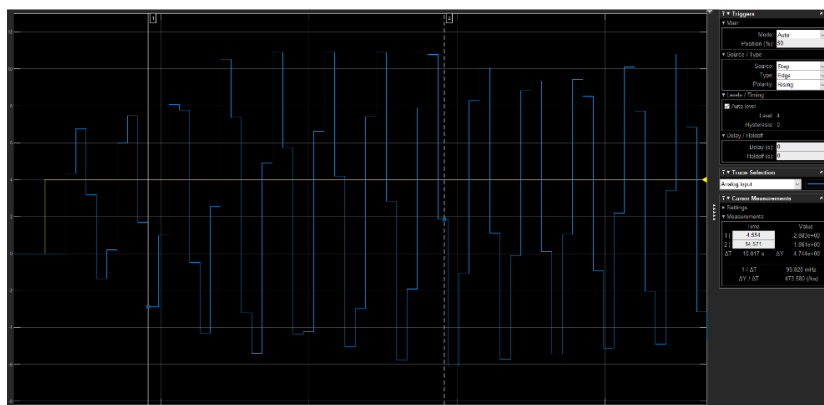


Figure 13 - Commande LQ ($p=2.5$) appliquée sur le moteur