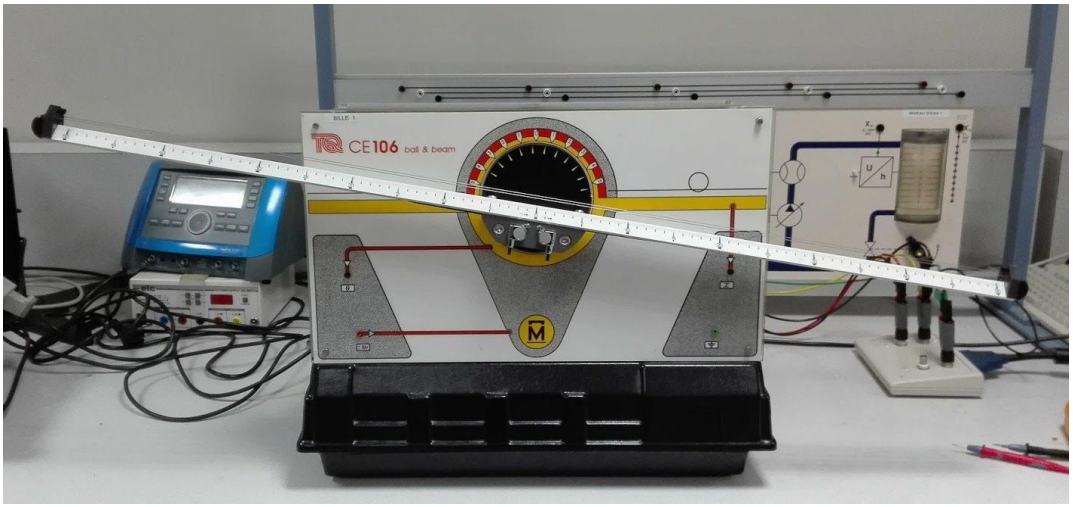
Rapport

Mini Projet d'Automatique

Maquette : Rail + Bille



Lucie Hernandez - luhernan@etud.insa-toulouse.fr
Julian Hoyos - hoyosrod@etud.insa-toulouse.fr

Sommaire :

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

INTRODUCTION

L'objectif de ce mini projet d'automatique est de commander ou réguler un système dont les caractéristiques internes sont au départ inconnues. Pour cela, une première étape consiste à identifier expérimentalement un modèle du système. Cela revient à l'étude d'un système dit "boîte noire" : on sollicite le système à l'aide de **Simulink Real Time** et on mesure les résultats en sortie. A partir des données obtenues, ils est possible, notamment grâce à la toolbox **Ident** de Matlab d'extraire un modèle du système, le but étant de trouver un modèle le plus fiable possible. C'est sur ce modèle que l'on peut ensuite travailler pour réguler le système, tout d'abord en concevant des lois de commandes grâce aux outils de simulation puis en les testant sur le système réel pour les valider.

Nous avons choisi de d'étudier la maquette ci-dessous :

Figure 1: Maquette bille + rail

Il s'agit d'un bras rotatif dont on peut modifier l'angle avec un moteur que l'on commande en tension. Le bras est constitué d'un rail métallique sur lequel on pose une bille, elle aussi métallique. Le contact de la bille avec le rail ferme un circuit, dont on peut mesurer la résistance et ainsi déduire la position de la bille sur le rail.
Le système est fortement non linéaire, il s'agit d'un système plus complexe qu'un simple servo-moteur dont l'angle correspondrait à l'angle du rail. Il possède une seule entrée, qui est la tension appliquée au moteur variant entre -10V et +10V, et deux sorties. La première sortie est la tension image de l'angle et la deuxième la tension image de la position de la bille. C'est un système totalement nouveau pour nous, puisque nous ne l'avons jamais rencontré lors des TP d'automatique à l'INSA.

Après une étape de prise en main de la maquette et l'identification de son modèle, notre objectif est de mettre en place une boucle de régulation de l'angle du rail puis une boucle de régulation de la position de la bille.

I- Prise en main de la maquette

Nous cherchons dans un premier à temps à comprendre de façon qualitative le fonctionnement du système, de façon à pouvoir être capable d'interpréter les résultats expérimentaux rencontrés dans la suite du projet. Nous cherchons notamment la plage dans laquelle le comportement du système peut être considéré comme linéaire et dans laquelle nous choisirons notre point de fonctionnement, ainsi que la zone "morte" du système, c'est à dire la plage de commande pour laquelle le moteur ne réagit pas à cause des frottements secs. Dans cette partie, nous n'utilisons pas la bille et nous concentrons sur les variations de l'angle du bras.

1) Etude de la réponse du système à un sinus

Le but de cette première manipulation est de comprendre "avec les mains" comment se comporte le système en boucle ouverte. Nous utilisons pour cela un signal d'entrée sinusoïdal de basse fréquence. Cela nous permet d'observer à l'oeil nu les variations de l'angle.
On remarque tout d'abord que sans signal d'entrée, celui-ci est initialement de -20° environ. De plus, le système mécanique faisant varier l'angle est composé d'un galet de forme ovale que le moteur fait tourner.
La tension d'entrée du moteur étant limitée à ± 10V, nous envoyons dans un premier temps un signal d'amplitude 10V, à l'aide des interfaces de Simulink Real Time (cf.figure 3). Nous observons que le bras oscille autour d'un angle inférieur à 10°. Le signal de sortie étant limité à 10V, il est impossible de visualiser les oscillations.
Nous essayons alors de faire osciller le système autour de 0°. Pour cela nous ajoutons une composante continue au signal d'entrée. Nous ajoutons un offset de 3V et réduisons l'amplitude du signal à 7V afin de ne pas dépasser les 10V lorsque le signal atteint son maximum. On observe alors que le système est déstabilisé car le sommet du galet est atteint, celui-ci effectue alors un tour complet et le comportement du bras est alors complètement non linéaire (cf. figure 2).

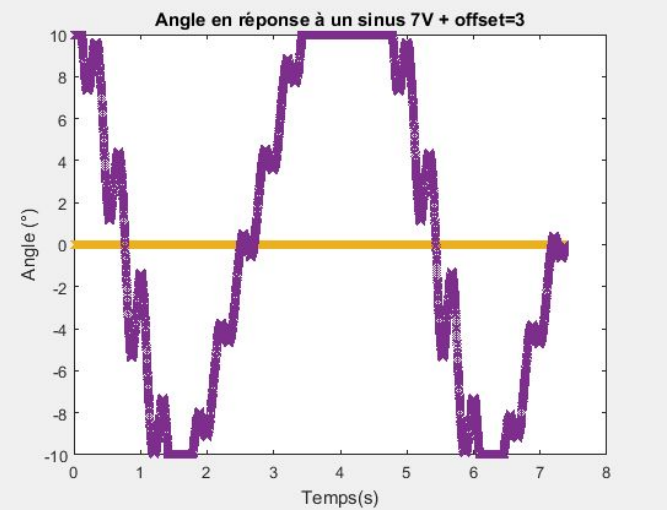


Figure 2: Variations de l'angle pour un sinus de 7V et un offset de 3V

Nous réduisons donc l'offset à 1V et augmentons l'amplitude à 9V. Cette fois-ci le système oscille bien autour de 0° (cf. figure 3). Nous déduisons donc de cette expérience qu'une valeur moyenne de 1V en entrée est nécessaire pour que l'angle soit centré autour de 0°.

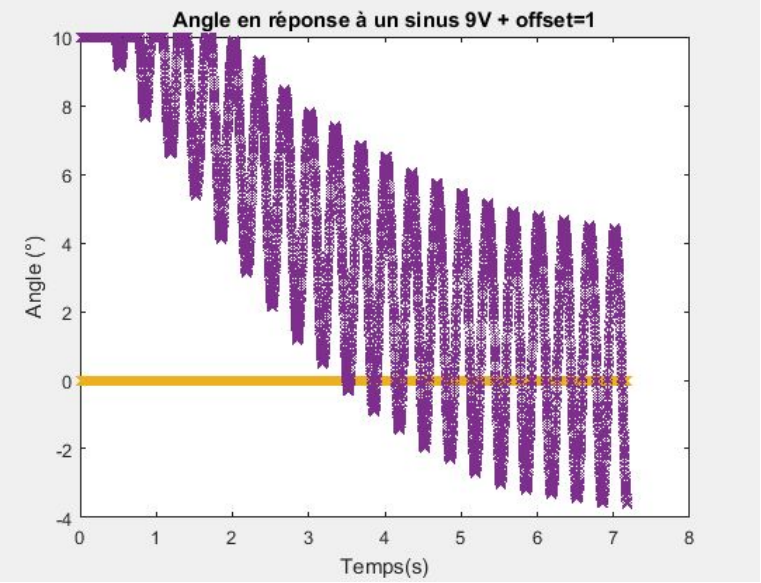


Figure 3: Variations de l'angle pour un sinus de 9V et un offset de 1V

2) Etude de la "zone morte" du système

Nous sollicitons le système avec un signal constant afin de déterminer les plages de tensions acceptables en termes de commande pour que le bras bouge. On remarque que pour des tensions comprises dans l'intervalle [-1,1V ; 2V] , le moteur ne réagit pas. C'est ce que nous allons appeler la zone morte. L'existence de cette zone s'explique par la présence de frottements secs qui s'opposent au mouvement du moteur.

II- Identification d'un modèle

Dans cette partie, nous cherchons à identifier un modèle du système. Pour cela deux approches sont possibles. L'approche classique, que nous avons utilisée jusqu'à présent dans les TP à l'INSA, consiste à observer les réponses indicielle et fréquentielle du système (réponse à un échelon, réponse à des signaux sinusoïdaux de fréquence variable …). Elle nécessite cependant des signaux d'excitation d'amplitudes importantes qui ne conviennent pas à notre système. En effet, celui-ci étant fortement non linéaire, les signaux de trop grande amplitude le déstabilisent et ne permettent pas d'obtenir des résultats exploitables.

Nous nous intéressons donc à la deuxième approche. Celle-ci consiste à utiliser des signaux d'excitation de faible amplitude très riches en fréquence et à utiliser les algorithmes d'identification de la toolbox Ident de Matlab pour extraire un modèle. Il faut donc dans un premier temps générer ces signaux. Pour cela nous allons générer une Séquence Binaire Pseudo Aléatoire (SBPA).

1) Réalisation d'une SBPA

Une SBPA est une série d'impulsions modulées aléatoirement en longueur. Celle-ci sera un multiple de la fréquence d'horloge. Le signal obtenu peut être assimilé à un bruit blanc discret, de valeur moyenne nulle et riche en fréquence. Ce signal ne modifie pas le point de fonctionnement du système et est très souvent utilisé dans le contexte d'identification de procédé.

Une SBPA est générée à partir de registres à décalage bouclés. Si N est le nombre de cellules dans le registre, la longueur de la séquence est alors de $(2^N - 1)$. La durée de la séquence est alors de $(2^N - 1).T_H$ où T_H est la période d'horloge utilisée dans les registres à décalage. A l'intérieur de la séquence, la longueur des impulsions varie de façon aléatoire, entre $1T_H$ et $N.T_H$, mais sur le long terme, la séquence se répète de façon périodique.

Nous choisissons de générer une SBPA à partir d'un registre à 8 cellules (cf. figure 4) et fixons notre période d'horloge à 20 fois la période d'échantillonnage. Le signal d'horloge joue le rôle de signal "enable".Nous pouvons agir sur les paramètres Gain A et Offset afin de faire varier l'amplitude du signal et sa valeur moyenne.

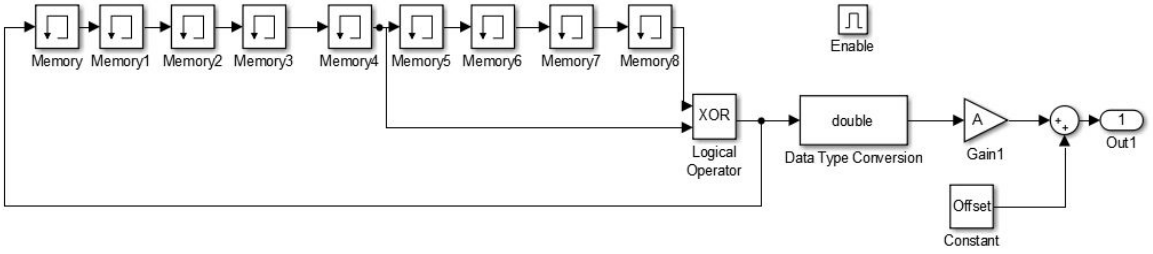


Figure 4: Schéma bloc du registre à décalage

En simulation nous obtenons le signal présenté en figure 5, en fixant $A = 20$ et $Offset = -10$. On observe qu'à l'échelle d'un seconde, la durée des impulsions (en rouge) semble bien varier de façon aléatoire et qu'elle est multiple de la période d'horloge (en bleu).

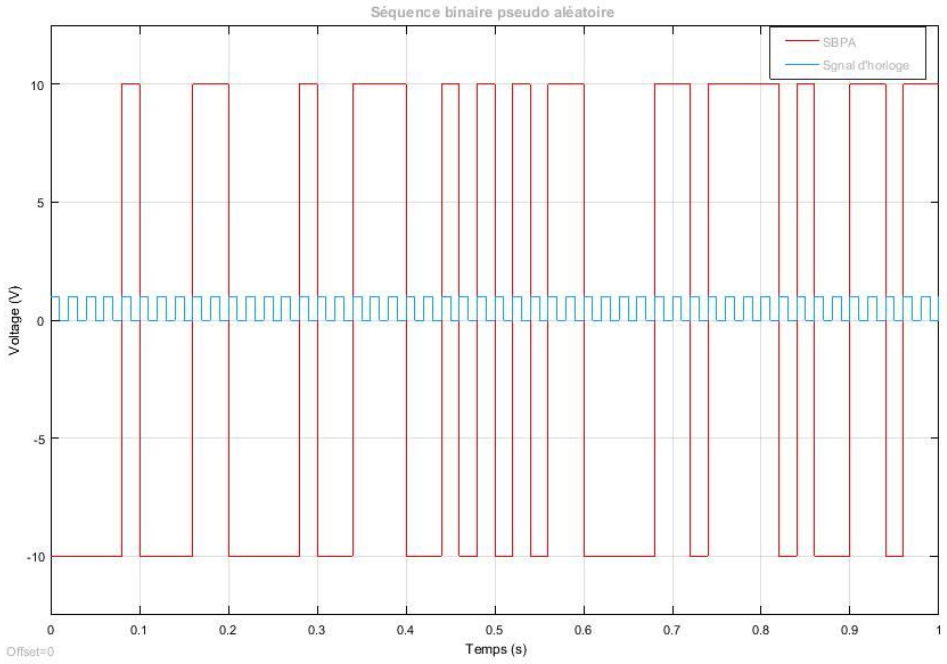


Figure 5: Séquence SBPA obtenue en simulation

2) Etude de la réponse du système au signal SBPA

Nous rajoutons un offset de 1V à la séquence obtenue précédemment (en fixant $A = 14$ et $Offset = -6$, afin de compenser le décalage au niveau de l'angle observé lors de la prise en main. La figure 6 représente la réaction du système (en bleu) à ce signal d'excitation (rouge).

Nous réalisons un deuxième test avec une SBPA légèrement différente ($A = 18$ et $Offset = -8$). Nous allons utiliser la toolbox **Ident** afin d'exploiter ces résultats.

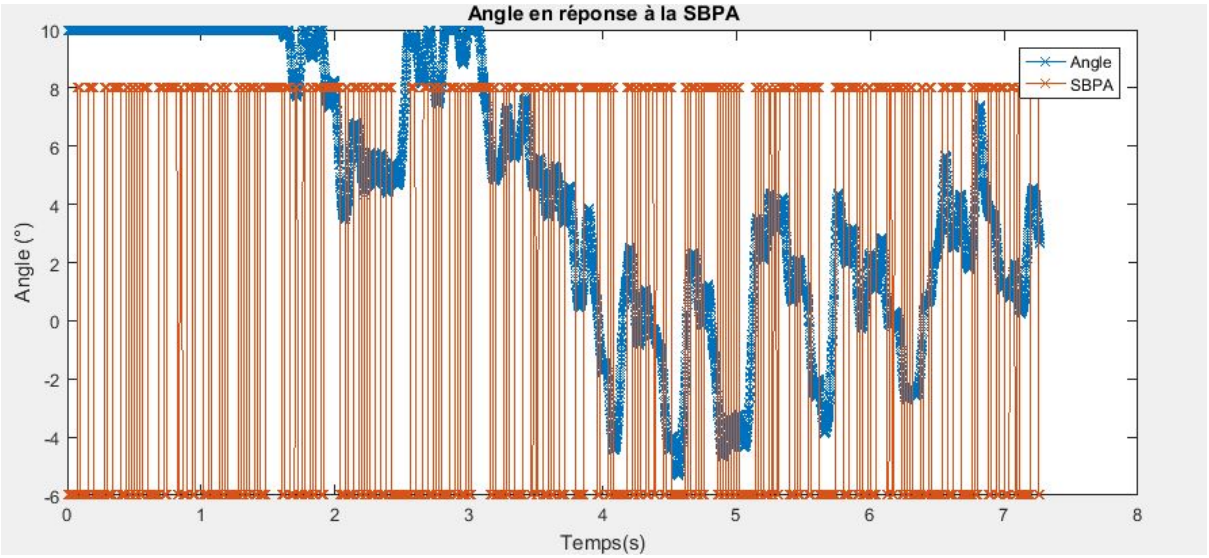


Figure 6: Réponse du système à la SBPA

3) Traitement des jeux de données

Nous exportons les jeux de données obtenus vers Ident. Afin de pouvoir les exploiter, il est nécessaire de les traiter, car un usage de ces valeurs brutes ne permet pas d'extraire un modèle précis. Nous réduisons donc dans un premier temps les jeux de données aux plages non saturées : nous retirons les parties du signal pour lesquels la sortie atteint -10V ou +10V. Nous enlevons ensuite la composante continue, car ce qui nous intéresse c'est le modèle représentant les variations du système autour de son point de fonctionnement.

4) Extraction des différents modèles, choix du modèle

Nous obtenons ainsi deux jeux de données, Test1 et Test2. Ces données sont des données d'estimation. A partir de chaque jeu de données, il est possible d'extraire différents modèles car plusieurs algorithmes peuvent être utilisés en fonction du type de modèle que l'on veut obtenir (fonction de transfert, ARX, ARMAX, Output Error etc.). Nous ne gardons pour chaque test que les deux modèles pour lequel le taux de correspondance est le plus élevé. L'interface Ident est présentée sur la figure 7. Les jeux de données sont à gauche et les modèles obtenus à droite. Les modèles "oe331" et "tf1" sont les modèles correspondent au Test1 et les modèles "oe221" et "tf2" au Test2.

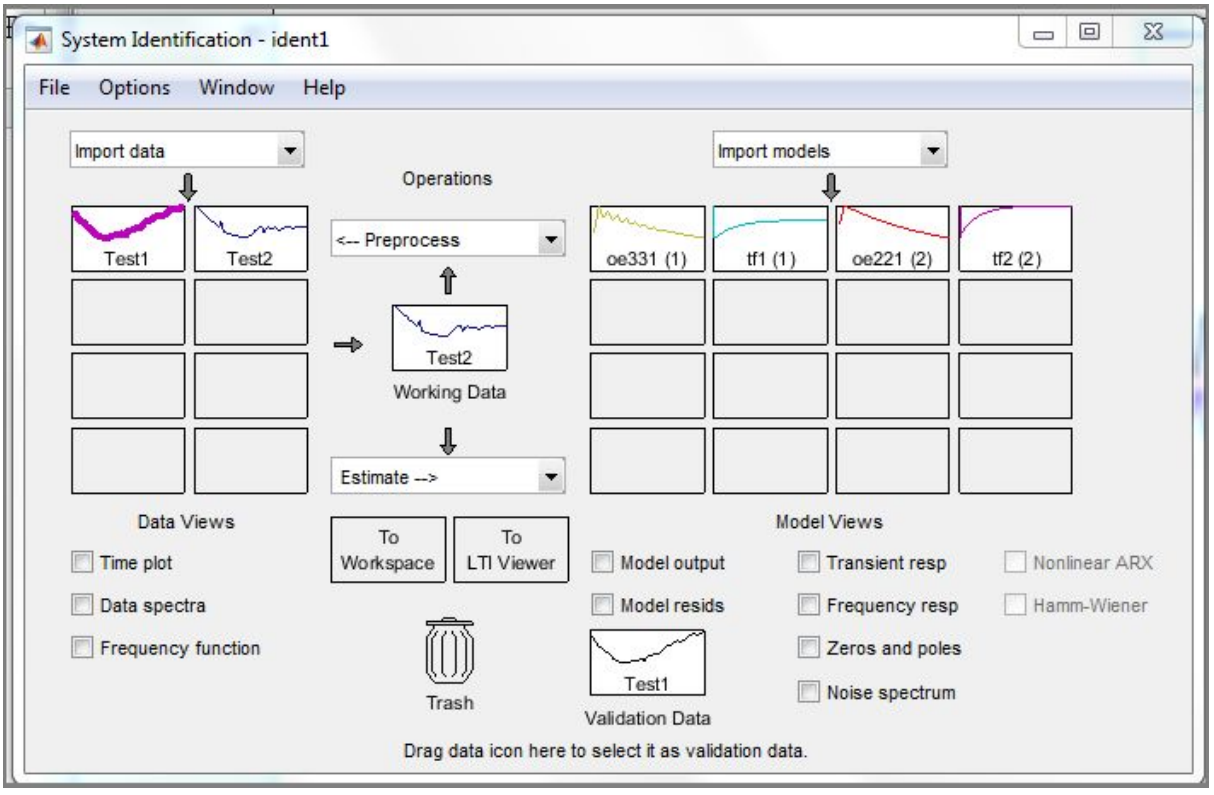


Figure 7: Interface Ident contenant les jeux de données et les différents modèles

Afin cependant de vérifier et ensuite choisir un modèle, il est nécessaire de le comparer le comparer par rapport à un jeu de donnée différent des données d'estimation. Ainsi, pour chaque modèle obtenu à partir du Test1, le Test2 servira de jeu de validation et

vis-versa.Les résultats obtenus lors de l'étape de validation sont présentés sur les figures 8 et 9.

On observe que les modèles "tf2" et "oe221" ont les plus grands taux de compatibilité, avec respectivement 35.52% et 41.37%. En se basant seulement sur ces grandeurs, on aurait tendance à choisir le modèle "oe221". On remarque toutefois que la réponse du modèle "tf2" oscille moins et se rapproche plus selon ce critère de la courbe de validation. De plus, limiter les oscillation est dans notre intérêt en ce qui concerne l'étape de régulation. Ainsi c'est ce modèle que nous choisissons. Il correspond à une fonction de transfert d'ordre 3 dont les coefficients sont les suivants :

```
  From input "u1" to output "y1":
             -1.814e04
  --------------------------------
  s^3 + 53.24 s^2 + 2204 s + 903.2
Name: tf2 (2)
Continuous-time identified transfer function.
```

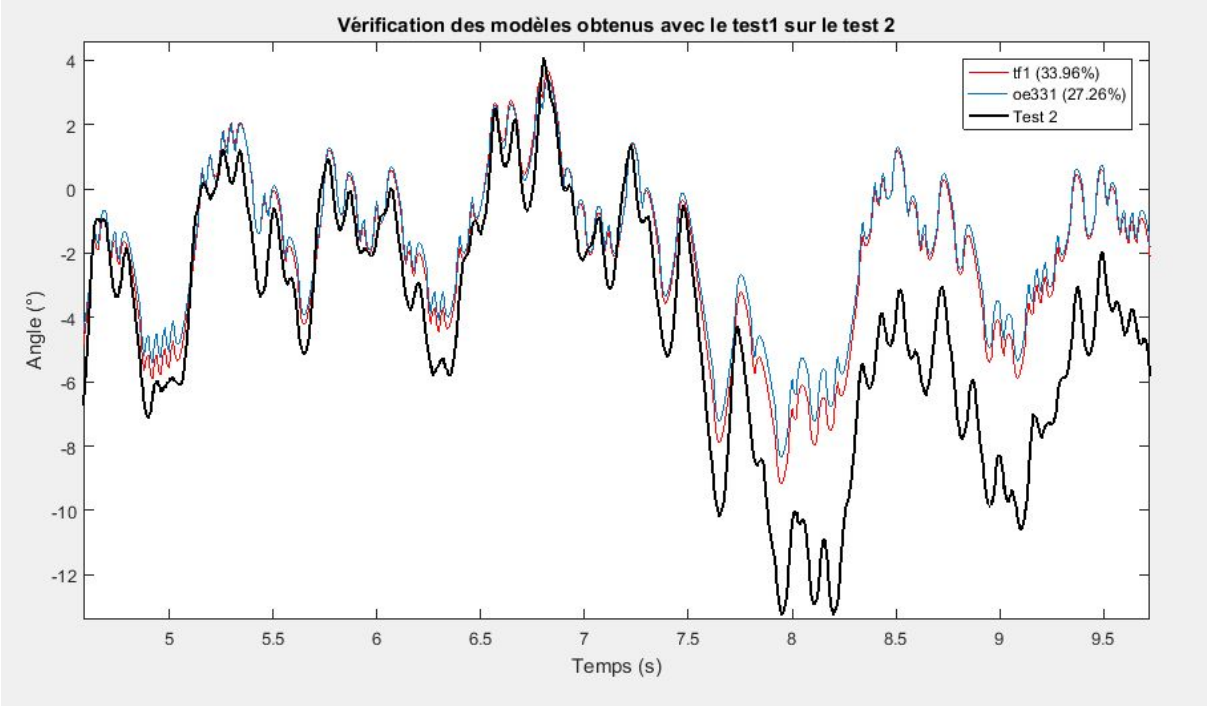


Figure 8: Validation des modèles "tf1" et "oe331"

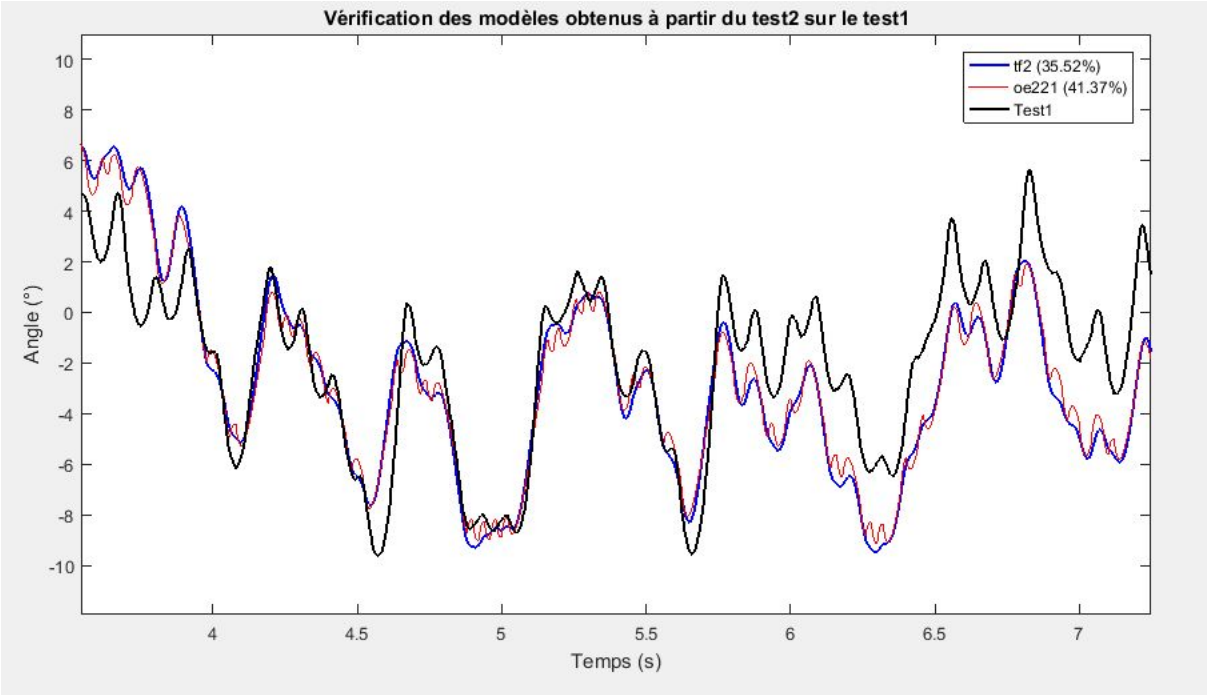


Figure 9: Validation des modèles “tf2” et “oe221”

Après avoir choisi le modèle avec lequel nous allons travailler, nous passons à l’étape de régulation.

III- Régulation de l’angle du bras

A partir du modèle choisi précédemment, nous procédons à la mise en place d’une commande qui permet de réguler l’angle du bras. D’abord un simple correcteur proportionnel est proposé afin de vérifier si un gain proportionnel peut suffir comme commande fiable. A priori, en regardant la fonction de transfert expérimentale, nous pouvons déduire qu’un gain proportionnel ne sera pas suffisant et que le système aura sûrement une erreur statique en régime permanent.

1) Correcteur P

Nous cherchons tout d’abord de manière expérimentale les valeurs que le gain du correcteur K_p peut prendre sans déstabiliser le système en boucle fermée.
Nous vérifions ensuite ces résultats grâce à l’outil **Sisotool** de Matlab. Le même résultat est trouvé :le gain limite pour ne pas rendre le système instable correspond à celui du résultat théorique.
En partant d’un gain qui se trouve dans l’intervalle des valeurs acceptables, plusieurs tests sont faits pour comprendre le comportement du système et pour trouver le gain qui donne le meilleur résultat en termes de temps d’établissement et d’erreur statique.

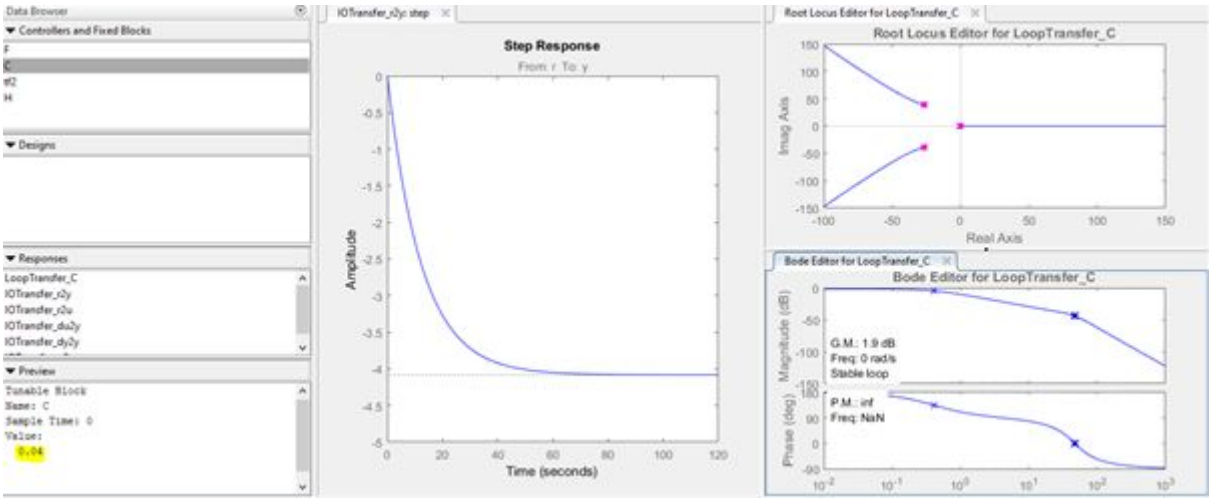


Figure 10: Vérification de K_p limite avec Sisotool

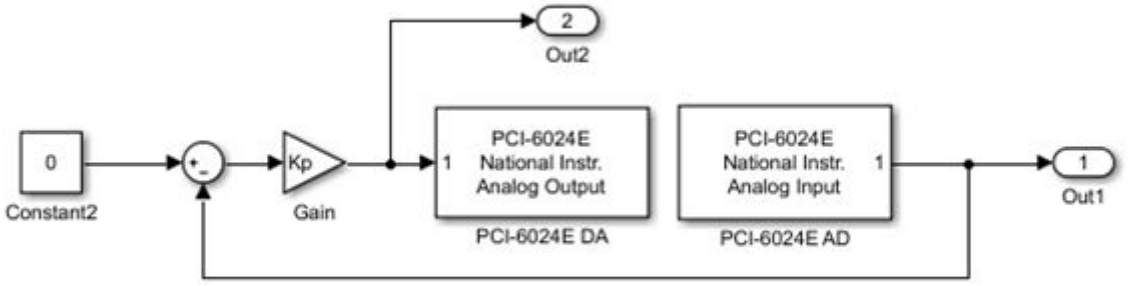


Figure 11: Schéma bloc du système avec correcteur proportionnel

Avec un gain $K_p = -5$, le système se comporte comme sur la figure 11.

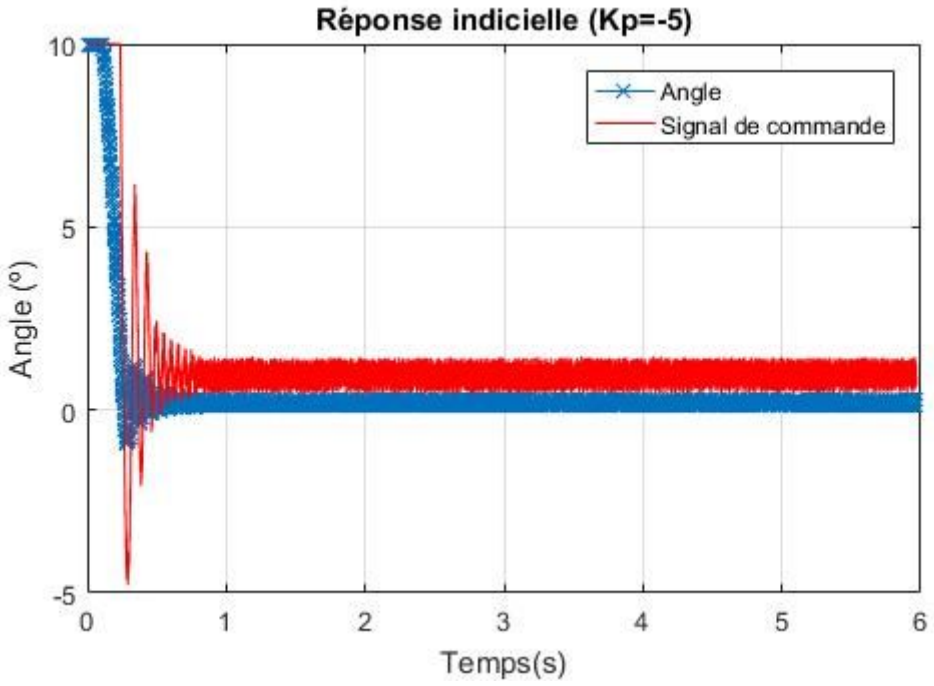


Figure 12: Réponse indicielle du système avec $K_p = -5$

On mesure l'erreur statique de l'angle, qui se stabilise à peu près à 0.8º et pas autour de l'angle de référence qui dans ce cas vaut 0º. Donc, nous avons bien trouvé le résultat attendu, effectivement, il existe une erreur statique du système.
En analysant les caractéristiques de la réponse indicielle, nous observons une réponse avec un nombre considérable d'oscillations et un dépassement de presque 12%, ce qui n'est pas désirable. Cependant, le temps d'établissement est de 0.23s, ce qui est assez rapide pour notre système.

Nous essayons de diminuer les oscillations et le dépassement. Un deuxième test est mené à bien, cette fois-ci avec un gain $K_p = -3$, plus petit que le précédent et nous comparons les deux réponses (cf. figure 13).

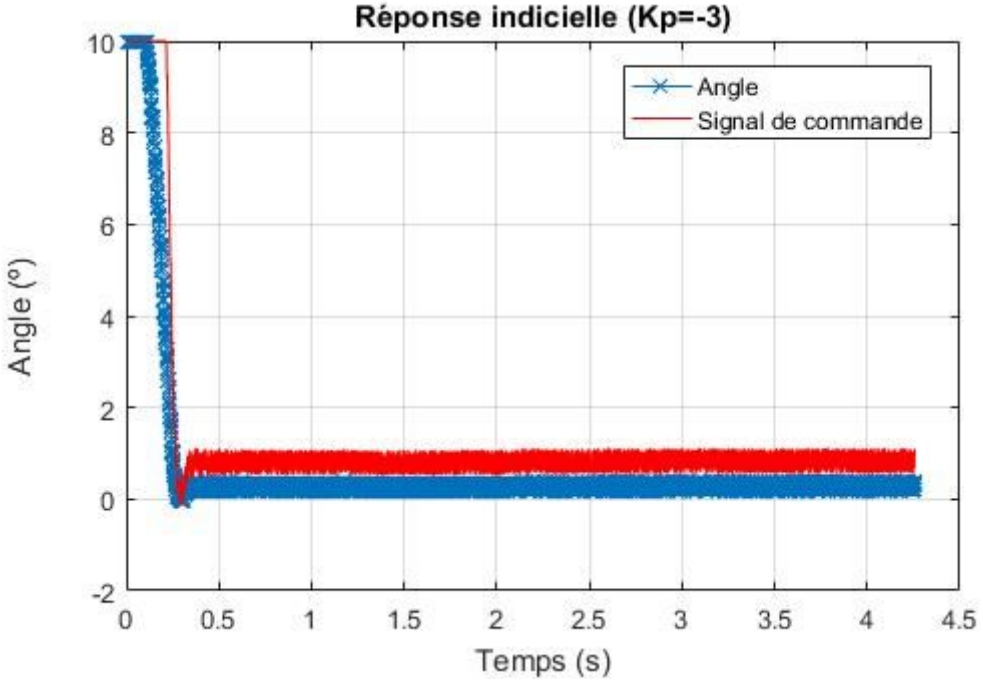


Figure 13: Réponse indicielle du système avec $K_p = -5$

Dans ce cas, le temps d'établissement du système à été un peu dégradé (0.35 s) par rapport à celui du test précédent, mais nous avons beaucoup diminué le dépassement à 8.6% et les oscillations ont disparu. Néanmoins, le problème de l'erreur statique est toujours présent.
Afin de nous débarrasser de ce problème, une commande proportionnelle et intégrale est ensuite proposée.

2) Correcteur PI

L'ajout d'une action intégrale nous aidera à faire disparaître l'erreur statique, puisque l'action intégrale est chargée de prendre en compte toutes les valeurs d'erreurs précédentes et d'agir sur la somme de ces erreurs. Ainsi même une erreur petite, que le correcteur proportionnel ne peut pas corriger suffisamment agir sera corrigée par l'action intégrale qui amènera le système à l'angle de référence.

Grâce à l'outil Sisotool de Matlab et en utilisant la fonction de transfert expérimentale, il est possible de trouver une première approximation des gains K_p et K_i. L'objectif est d'obtenir une réponse indicielle sans dépassement, sans oscillations et la plus rapide possible. La robustesse du système, qu'il est possible d'observer dans le diagramme de Bode (cf. figure 14), est également un critère à prendre en compte.

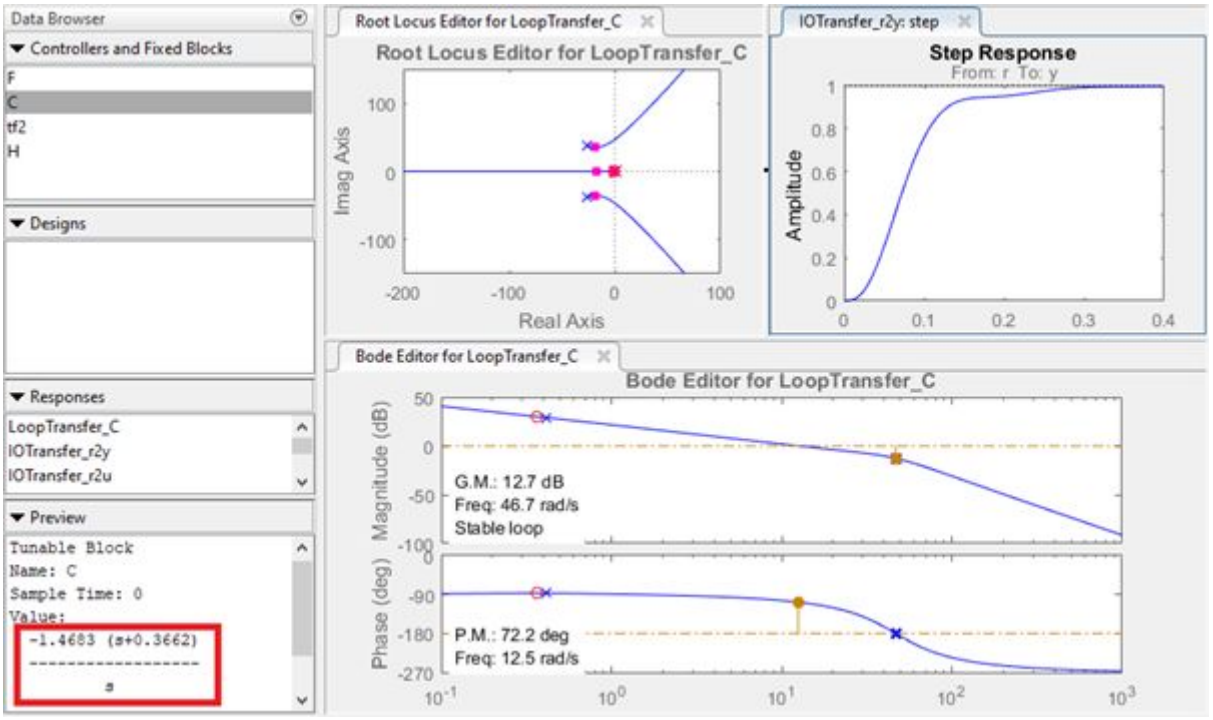


Figure 14: Design du correcteur PI au moyen de Sisotool

A partir des valeurs extraites de la simulation, plusieurs tests ont été faits en vue de trouver celles qui donnent comme résultat la meilleure réponse. Dans le tableau suivant toutes les caractéristiques et commentaires des différents tests sont affichés.

K_p	K_i	Temps d'établissement (s)	Dépassement	Commentaires
-0,8	-2	6	5%	Réponse lente, il prend assez de temps pour arriver à l'erreur statique nulle. L'augmentation du gain intégral est considérée.
-1,2	-2,5	1,5	3%	La rapidité a été améliorée, ainsi que le dépassement. Par contre, nous cherchons un meilleur résultat. L'augmentation des deux gains est considérée

-6	-5	1	15%	Réponse très rapide, mais avec un dépassement assez considérable. La réduction de l'effet proportionnel est envisageable.
-2,5	-5	1.2	2%	Dépassement presque nul, rapidité acceptable. Meilleur résultat obtenu.

Finalement, les meilleures valeurs du correcteur PI sont les suivantes : $K_p = -2,5$ et $K_i = -5$. En traçant la réponse indicielle du système, nous obtenons la figure 15.

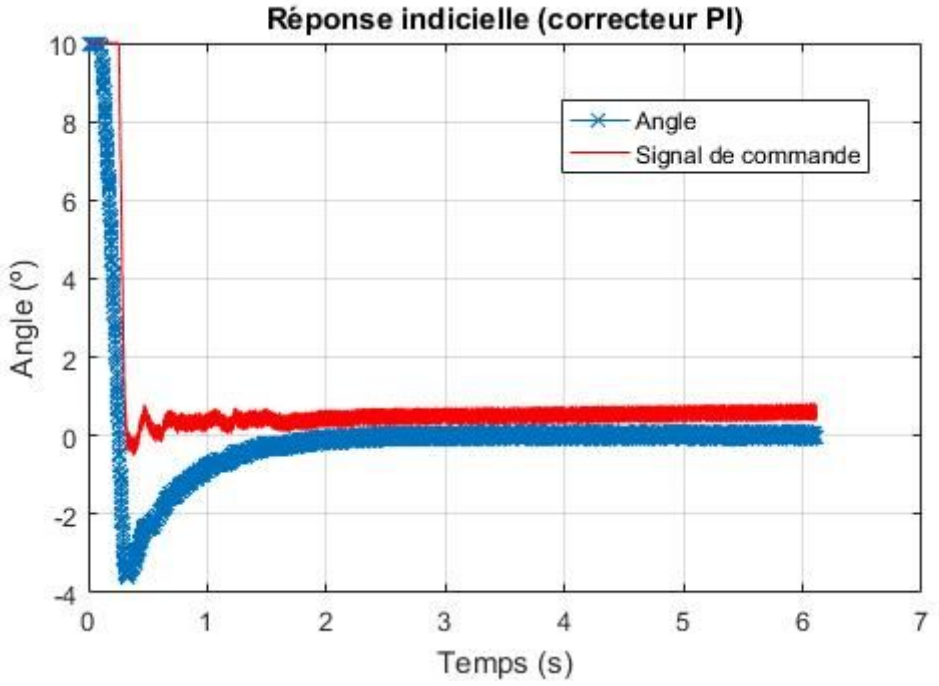


Figure 15: Réponse indicielle du système avec correcteur PI

Nous constatons qu'il n'y a pas d'erreur statique, l'angle se stabilise autour de 0°, même si ce n'est pas la valeur exacte due à la précision des capteurs et à la zone morte liée au moteur. Il est possible d'observer l'action intégrale du correcteur qui fait que le signal de commande monte petit à petit. Par rapport au correcteur proportionnel, nous avons gagné en précision mais le temps d'établissement a été dégradé.

L'implémentation d'une composante dérivée dans le correcteur n'est pas envisagée puisque le correcteur PI donne au système une réponse indicielle très convenable.
Après avoir fait la régulation du système monovariable, nous ajoutons la position de la bille comme deuxième entrée du système.

IV- Régulation de la position de la bille par PID

Une fois la régulation de l'angle réalisée, nous cherchons à réguler la position de la bille sur le bras. Pour cela, nous allons considérer que la fonction de transfert entre la position de la bille et l'angle du bras est de la forme $H(s) = \frac{K}{s^2}$.
En effet, on peut partir des équations mécaniques pour trouver la relation entre la position et l'angle :

$$sin\,\theta.m.g = m.a\,.$$

où *m* est la masse de la bille, θ l'angle du bras, *g* la gravité et *a* l'accélération de la bille dans la direction du mouvement. En prenant *x* la position de la bille, l'équation précédente devient :

$$sin\theta.g = \ddot{x}$$

En sachant que pour des petits angles la relation est valable et en faisant la transformé de Laplace, nous obtenons :

$$\theta.g = s^2.X$$

D'où $H(s) = \frac{K}{s^2}$.

Nous considérerons de plus que la régulation de l'angle se fait très rapidement devant la régulation de la position de la bille. Nous pouvons alors approximer tout le système entre la tension d'entrée et la position de la bille par H(s) (cf. figure17).

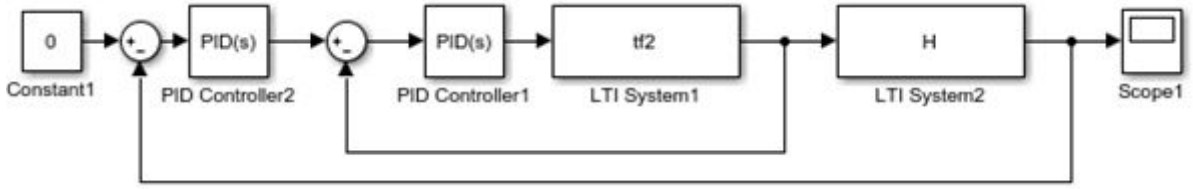


Figure 16: Schéma bloc du système bouclé complet

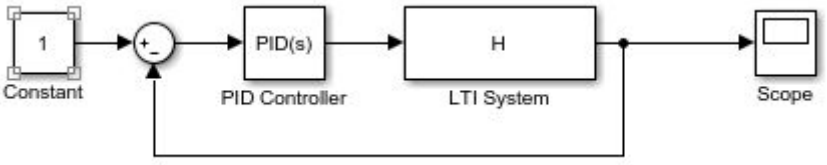


Figure 17: Schéma bloc du système bouclé réduit

Nous cherchons à déterminer expérimentalement la valeur du gain K. Pour cela, nous bouclons le système réel sur lui même avec un gain unitaire afin d'en étudier les auto-oscillations. Nous mesurons une fréquence d'oscillation de 0,1864 Hz. La relation entre la fréquence et K est la suivante : $K = (2\pi.f_o)^2$.

On obtient donc un gain K égal à 1,372. Nous pouvons alors tester en simulation la réponse du système en boucle fermée avec ce gain afin de vérifier que l'on a bien le même comportement. La figure 18 représente le résultat de la simulation. On retrouve une fréquence d'auto-oscillation de 0,189 Hz ce qui est cohérent.

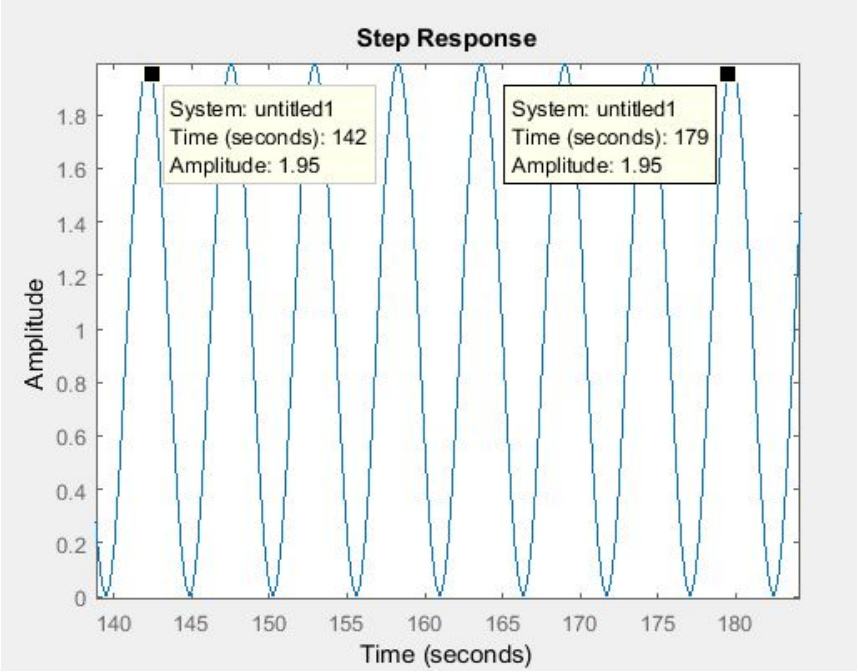


Figure 18: Auto-oscillations du système modélisé

D'abord en utilisant l'outil **PID Tunning** de Sisotool, nous trouvons les valeurs des gains de départ du correcteur PID.

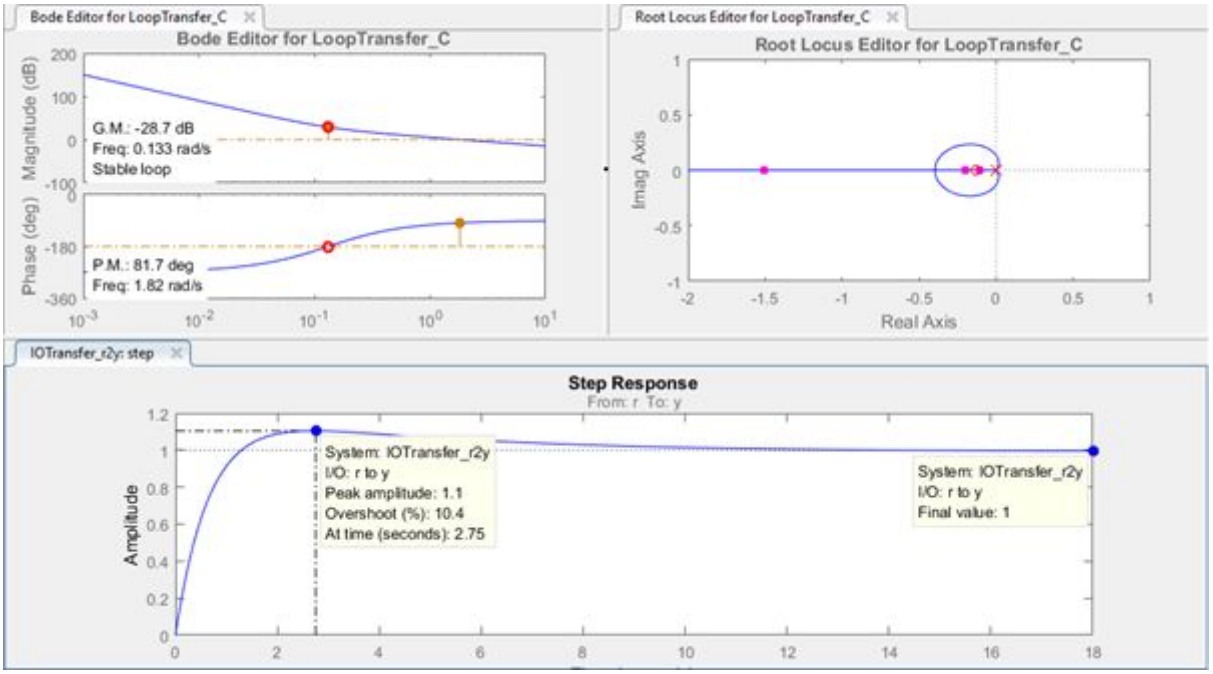


Figure 19: Réglage du correcteur PID avec Sisotool

Les valeurs initiales du correcteur que nous choisissons sont les suivantes :

$$K_d = 2,71 \qquad K_p = 0,639 \quad \text{et} \quad K_i = 0,00375$$

Pour l'implémentation en réel du correcteur de la deuxième boucle, nous gardons le correcteur PI trouvé dans la section précédent et nous ajoutons le deuxième correcteur qui agira sur le système de la bille afin de réguler sa position.

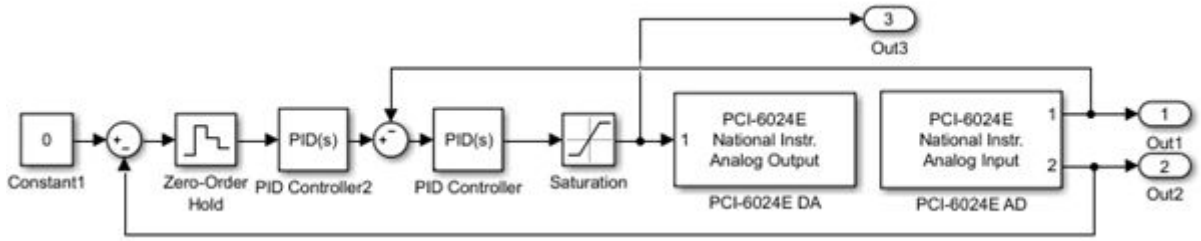


Figure 20: Schéma bloc de la régulation du système réel complet sous Simulink RT

A partir des valeurs initiales des gains du correcteur, nous faisons varier petit à petit les coefficients afin de trouver la meilleure réponse indicielle du système. Les résultats de ces essais sont consignés dans le tableau récapitulatif suivant.

K_d	K_p	K_i	Commentaire
2,71	0,63	0,037	Réponse assez réactive, pas trop robuste à de grande perturbations. La réduction de l'effet dérivatif est considérée.
0,58	0,13	0,008	Plus robuste par rapport à celle d'avant, mais trop lent. L'augmentation de la constante intégrale est considérée.
0,80	0,18	0,016	Réponse très robuste mais pas trop précise. Il faudrait augmenter encore plus le gain intégral.
0,80	0,18	0,107	Toujours robuste et plus précise, mais la réponse prend trop de temps pour arriver à l'erreur statique nulle. L'augmentation du gain proportionnel est considérée.
0,80	0,85	0,050	Réponse robuste, précise et rapide. La meilleure réponse trouvée.

La réponse indicielle du système complet en boucle fermée avec le meilleur correcteur trouvé est affichée ci-après (cf. figure 21).

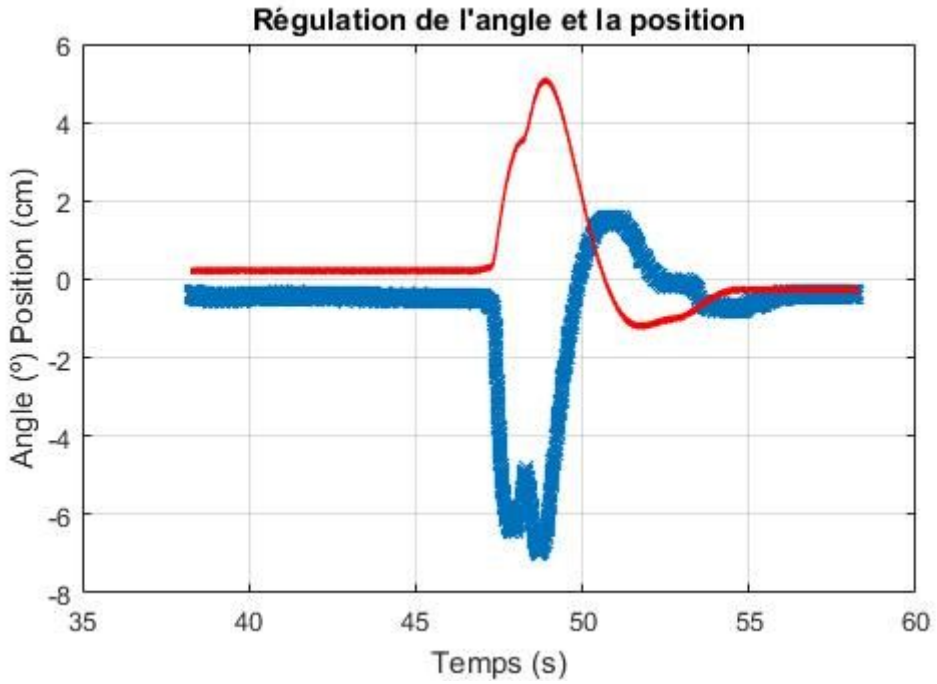


Figure 21: Réponse indicielle du système bouclé avec correcteur PID

Finalement, nous obtenons une régulation assez propre, rapide et précise qui est capable de rejeter les perturbations et de réguler le système à partir de n'importe quel point de départ. Même si les conditions initiales du système sont un peu éloignées du point d'équilibre, le système arrive à 0º en angle et autour du centre du rail en position.

V- Régulation de la position de la bille par commande LQR

Afin d'expérimenter un autre type de correcteur, nous décidons de réaliser une commande par retour d'état à l'aide de la méthode LQR.

1) Choix du modèle d'état

Il faut dans un premier temps trouver le modèle d'état correspondant au système. Nous considérons toujours la régulation de l'angle comme transparente du point de vue de la position, ce qui signifie qu'il nous suffit de trouver un modèle d'état pour la fonction de transfert H(s).
Nous sommes tombés dans le piège de la facilité et avons tout d'abord utilisé la commande $H_{ss} = ss(H)$. Cependant la représentation d'état obtenue a perdu tout sens physique et il nous est impossible de lier ses états aux grandeurs physiques que nous mesurons (angle et position). Nous calculons donc la représentation d'état à la main.
On pose :

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} u_x \\ \dot{u_x} \end{bmatrix}$$

avec u_x la position de la bille qui es la sortie du système, et u_θ l'angle du bras, qui est aussi l'entrée. La représentation d'état est la suivante :

$$\dot{X} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} \dot{u_x} \\ \ddot{u_x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} . \begin{bmatrix} u_x \\ \dot{u_x} \end{bmatrix} + \begin{bmatrix} 0 \\ K \end{bmatrix} .u_\theta \qquad y = \begin{bmatrix} 1 & 0 \end{bmatrix} . \begin{bmatrix} u_x \\ \dot{u_x} \end{bmatrix}$$

2) Test avec différents gains + schéma bloc

La commande correspondant au retour d'état est :

$$u = K_{lqr}.\varepsilon = K_{lqr}.(X_{ref} - X)$$

$$u = K_{lqr} \begin{bmatrix} x_{1ref} - x_1 \\ x_{2ref} - x_2 \end{bmatrix} = K_{lqr} \begin{bmatrix} x_{1ref} - u_x \\ x_{2ref} - \dot{u_x} \end{bmatrix} = K_{lqr} \begin{bmatrix} x_{1ref} - y \\ x_{2ref} - \int u_\theta . K \end{bmatrix}$$

A partir de ces équations, nous construisons le schéma bloc correspondant :

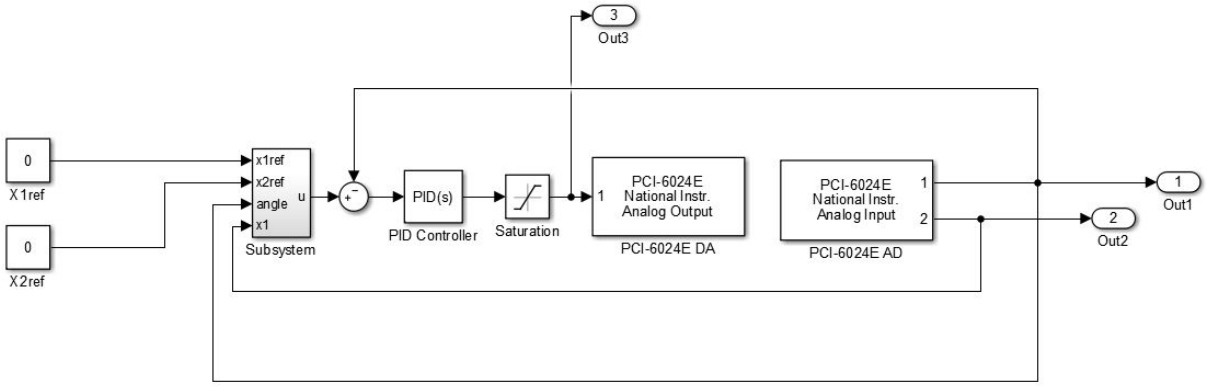


Figure 22: Schéma bloc du retour d'état

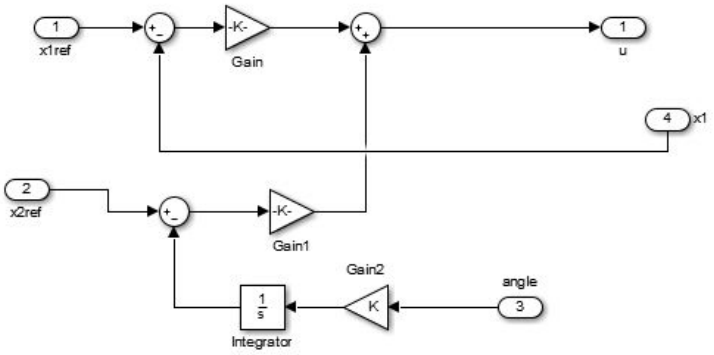


Figure 23: Vu sous le masque de la partie "Subsystem"

Le gain de retour est calculé à l'aide de la fonction lqr(A,B,Q,R) de Matlab. La matrice Q est une matrice carrée de dimension deux que nous choisissons diagonale. Nous pouvons agir sur la valeur de ses coefficients diagonaux q_1 et q_2. R est dans notre cas un scalaire que nous choisissons égal à 1.

Afin d'avoir une première idée de la réponse du système avec ce type de commande, nous réalisons un premier test avec $q_1 = q_2 = 1$. La réponse du système (cf. figure 24) est très

lente, on observe que la commande (**vert**) est elle faible et que par conséquent la position (**rouge**) de la bille n'est pas régulée.

Nous testons ensuite l'effet d'un gain plus important pour l'un ou l'autre des états. Lorsque l'on favorise le critère vitesse en fixant $q_2 = 10$ ($q_1 = 1$), on observe de grandes oscillations, car c'est la vitesse à qui l'on donne plus de poids dans notre correcteur. Ce comportement ne nous intéresse pas. Nous inversons alors et fixons $q_1 = 10$ ($q_2 = 1$). Cette fois, on observe moins d'oscillations (cf.figure 25). Cependant, on observe une erreur statique qu'il ne devrait pas y avoir, du fait de la nature doublement intégratrice du système. A chaque fois que l'on perturbe le système en faisant bouger la bille à la main, celui-ci commence à réguler mais "s'arrête" dès que l'angle du bras (**en bleu**) est nul.

Après de nombreux tests et beaucoup d'incompréhension, nous avons supposé que, puisque nous donnons en consigne une position mais aussi une vitesse nulle, le système stoppe sa régulation quand la vitesse de la bille est nulle, au dépend de sa position.

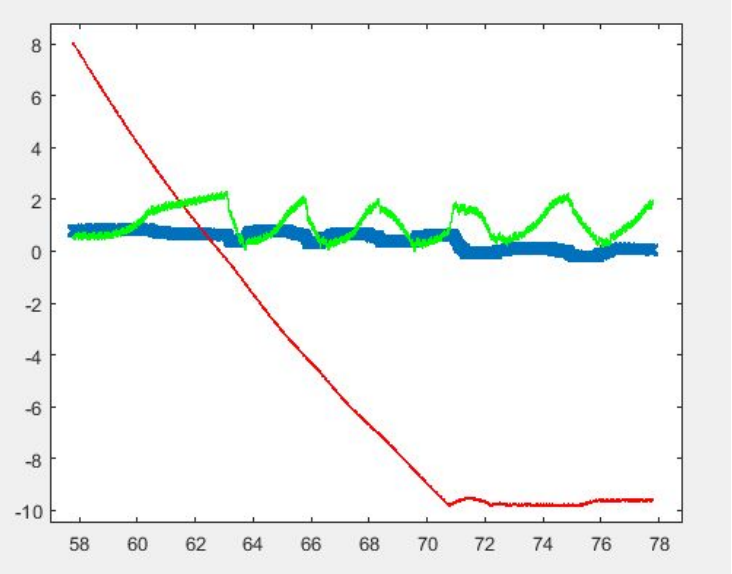


Figure 24: Réponse du système en boucle fermée ($q_1 = q_2 = 1$)

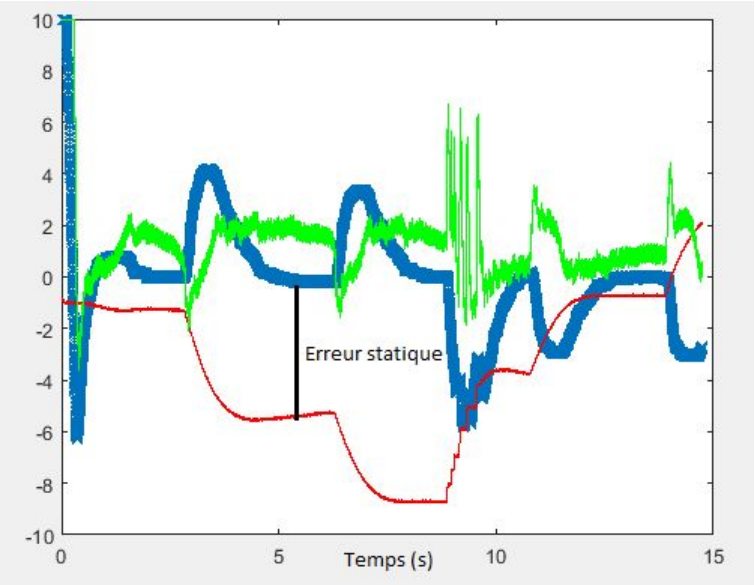


Figure 25: Réponse du système en boucle fermée ($q_1 = 10, q_2 = 1$)

Afin d'essayer de réduire l'erreur statique observée à chaque fois, nous décidons d'ajouter une action intégrale au retour d'état.

3) Ajout de l'action intégrale

Nous ajoutons l'action intégrale sur la position de la bille. Nous la rajoutons sur le schéma bloc de la figure 26.

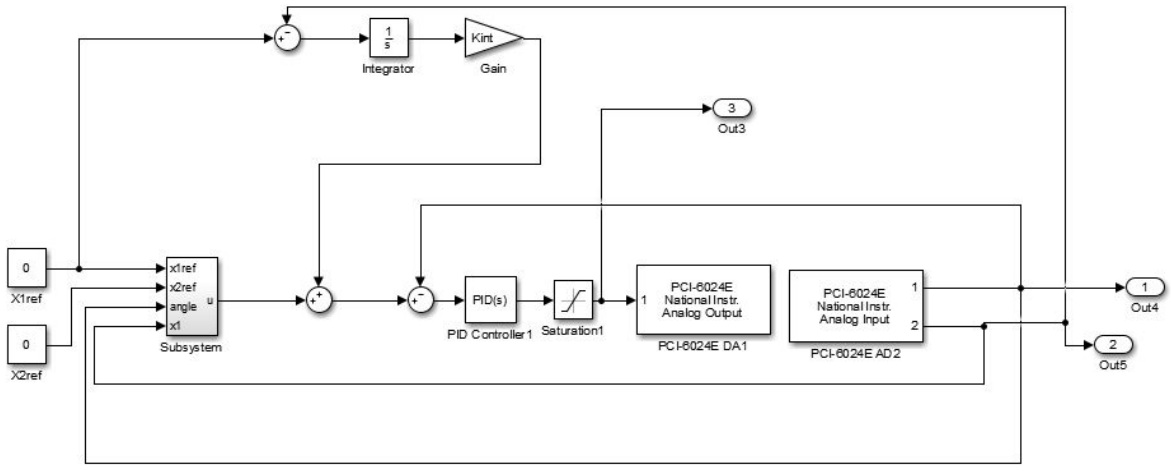


Figure 26: Schéma bloc avec action intégrale

Nous calculons le gain augmenté correspondant au nouveau système de la même manière que précédemment, en ajoutant un troisième coefficient dans la matrice Q, puisque nous ajoutons un troisième état au système qui correspond à l'intégrale de la position.

Nous réalisons plusieurs tests en faisant varier les trois paramètres q_1 , q_2 et q_3 . Le tableau ci-dessous résume nos observations, qui sont restées qualitatives par manque de temps.

Valeurs des paramètres	Qualité de la réponse
$q_1 = 10, q_2 = 1, q_3 = 1$	Quasiment même réponse que sans l'action intégrale, il faut mettre un plus gros coefficient. Erreur statique toujours présente.
$q_1 = 30, q_2 = 1, q_3 = 1$	Grandes oscillations, car gain et commande trop forts.
$q_1 = 10, q_2 = 5, q_3 = 1$	Le système réagit lentement, on n'observe pa vraiment de régulation, l'erreur statique est d'environ 10-20 cm.
$q_1 = 10, q_2 = 1, q_3 = 5$	L'erreur statique a diminué (5 cm environ).

Ces tests nous permettent de constater que nous ne pouvons pas trop augmenter le coefficient q_1 car il a un impact fort sur l'amplitude de la commande et donc celle des oscillations. De plus, on remarque qu'il faut augmenter de façon importante le coefficient associé à l'action intégrale pour en voir l'effet.

Nous obtenons le meilleur résultat de ce correcteur avec les coefficients $q_1 = 10, q_2 = 1, q_3 = 50$. La figure 27 représente la réponse observée. Il n'y a plus d'erreur statique, mise à part une faible erreur liée à l'inclinaison de la table sur laquelle le système est posé. Suite à une perturbation (à t=23s) et même des perturbations successives (à t=26s), on observe que le système réussit à positionner la bille à 0.

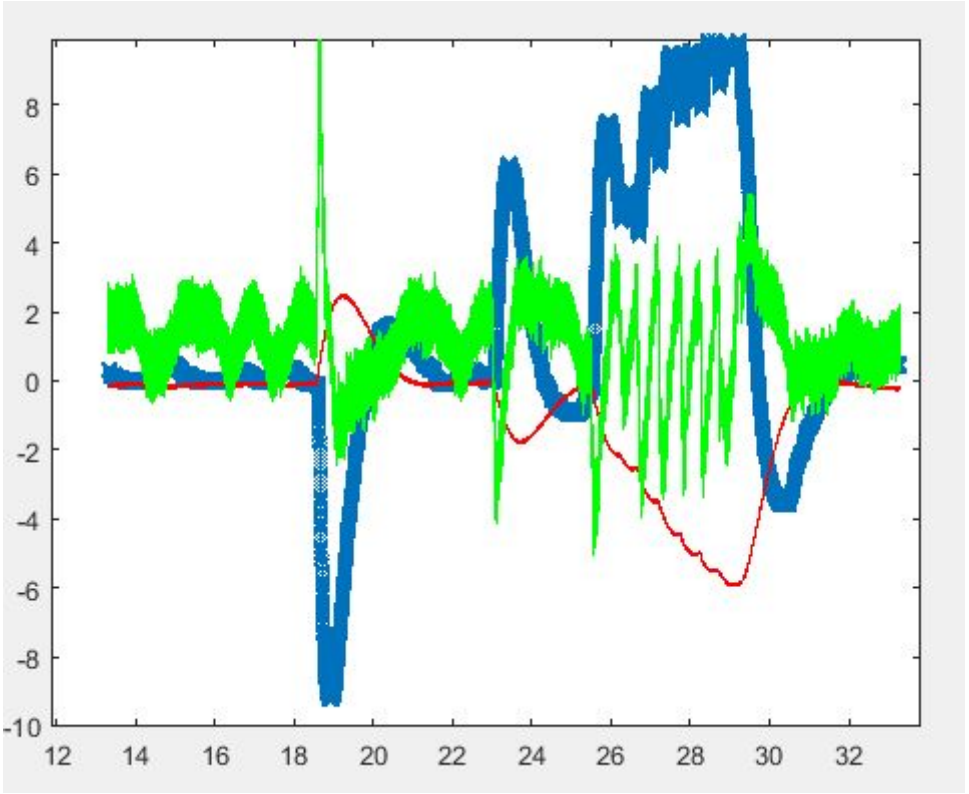


Figure 27: Réponse du système bouclé avec action intégrale ($q_1 = 10, q_2 = 1, q_3 = 50$)

Pour conclure concernant ce correcteur, nous notons qu'il n'est pas approprié à notre système. Il n'aurait pas dû nécessiter d'action intégrale. Nous avons tout de même consacré beaucoup de temps à réaliser des tests pour comprendre le comportement du système.

CONCLUSION

Le Mini Projet d'automatique a été pour nous l'occasion d'appréhender le raisonnement d'un automaticien dans sa globalité et sa complexité, en passant par la phase de modélisation jusqu'au choix et au design de lois de commande.

Lors de ce projet d'automatique, nous avons découvert une nouvelle approche permettant de modéliser un système, notamment un système dont on ignore tout. A l'aide d'un signal SBPA et de la toolbox **Ident** de Matlab, que nous utilisions pour la première fois, nous avons trouvé un modèle du système étudié.
Nous avons ensuite pu le réguler à l'aide dans un premier temps de correcteurs PI et PID, que nous sommes maintenant habitués à manipuler. Le correcteur PID a donné de très bons résultats, la réponse du système est fluide, sans à-coups, sans bruit et le système est robuste à de fortes perturbations.
Dans un second temps, nous avons essayé de réaliser une commande LQR par retour d'état, qui s'est avérée beaucoup moins efficace avec de fortes erreurs statiques. Nous avons rajouté une action intégrale qui nous a permis de réduire cette erreur statique, mais globalement ce correcteur est beaucoup moins efficace que le premier.
Lors de la conception des deux correcteurs, nous avons couplé l'utilisation de simulink et de simulink real time, afin d'anticiper les réponses du système par des simulations et d'y comparer ensuite les réponses du système obtenues expérimentalement. Cette méthode nous a été très utile lors de la conception du correcteur PID. Lors du design du correcteur LQR, nous avons la plupart du temps expérimenté directement sur le système réel, car les résultats divergeaient des simulations.

Ce projet nous a également permis de travailler en autonomie et nous a laissé pas mal de liberté, notamment dans nos choix de correcteurs, même si nous étions accompagnés tout au long.